Preprint
UCRL-ID-144763

Proceedings: Fourth Workshop on Mining Scientific Datasets

C. Kamath

This article was submitted to 4th Workshop on Mining Scientific Datasets, held in conjunction with the 7th International Conference on Knowledge Discovery and Data Mining, August 26-29, 2001, San Francisco, CA

July 24, 2001

U.S. Department of Energy

Lawrence
Livermore
National
Laboratory

This work was performed under the auspices of the United States Department of Energy by the University of California, Lawrence Livermore National Laboratory under contract No. W-7405-Eng-48.

This report has been reproduced directly from the best available copy.

Available electronically at http://www.doc.gov/bridge

Available for a processing fee to U.S. Department of Energy
And its contractors in paper from
U.S. Department of Energy
Office of Scientific and Technical Information
P.O. Box 62
Oak Ridge, TN 37831-0062
Telephone: (865) 576-8401
Facsimile: (865) 576-5728
E-mail: reports@adonis.osti.gov

Available for the sale to the public from
U.S. Department of Commerce
National Technical Information Service
5285 Port Royal Road
Springfield, VA 22161
Telephone: (800) 553-6847
Facsimile: (703) 605-6900
E-mail: orders@ntis.fedworld.gov
Online ordering: http://www.ntis.gov/ordering.htm

OR

Lawrence Livermore National Laboratory
Technical Information Department's Digital Library
http://www.llnl.gov/tid/Library.html

PROCEEDINGS

Fourth Workshop on Mining Scientific Datasets

in conjunction with the

7th ACM SIGKDD International Conference on
Knowledge Discovery and Data Mining

Edited by

Chandrika Kamath

Lawrence Livermore National Laboratory

August 26, 2001

San Francisco, CA

Fourth Workshop on Mining Scientific Datasets

August 26, 2001
San Francisco, CA

In conjunction with the

7th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining

WORKSHOP ORGANIZERS

Michael Burl, Jet Propulsion Laboratory, burl@aig.jpl.nasa.gov
Chandrika Kamath, Lawrence Livermore National Lab, kamath2@llnl.gov
Vipin Kumar, University of Minnesota, kumar@cs.umn.edu
Raju Namburu, Army Research Lab, raju@arl.mil

PROGRAM COMMITTEE

Dennis DeCoste, Jet Propulsion Laboratory, decoste@aig.jpl.nasa.gov
Sara Graves, University of Alabama at Huntsville, sgraves@itsc.uah.edu
Roberta Humphreys, University of Minnesota, roberta@isis.spa.umn.edu
Menas Kafatos, George Mason University, mkafatos@gmu.edu
Jacqueline Le Moigne, NASA Goddard, lemoigne@gsfc.nasa.gov
B.S. Manjunath, University of California at Santa Barbara, manj@ece.ucsb.edu
David Opitz, University of Montana, opitz@cs.umt.edu
Padhraic Smyth, University of California Irvine, smyth@ics.uci.edu
Roy Williams, Caltech, roy@cacr.caltech.edu

FOREWORD

Commercial applications of data mining in areas such as e-commerce, market-basket analysis, text-mining, and web-mining have taken on a central focus in the KDD community. However, there is a significant amount of innovative data mining work taking place in the context of scientific and engineering applications that is not well represented in the mainstream KDD conferences. For example, scientific data mining techniques are being developed and applied to diverse fields such as remote sensing, physics, chemistry, biology, astronomy, structural mechanics, computational fluid dynamics etc. In these areas, data mining frequently complements and enhances existing analysis methods based on statistics, exploratory data analysis, and domain-specific approaches.

On the surface, it may appear that data from one scientific field, say genomics, is very different from another field, such as physics. However, despite their diversity, there is much that is common across the mining of scientific and engineering data. For example, techniques used to identify objects in images are very similar, regardless of whether the images came from a remote sensing application, a physics experiment, an astronomy observation, or a medical study. Further, with data mining being applied to new types of data, such as mesh data from scientific simulations, there is the opportunity to apply and extend data mining to new scientific domains.

This one-day workshop brings together data miners analyzing science data and scientists from diverse fields to share their experiences, learn how techniques developed in one field can be applied in another, and better understand some of the newer techniques being developed in the KDD community. This is the fourth workshop on the topic of Mining Scientific Data sets; for information on earlier workshops, see http://www.ahpcrc.org/conferences/. This workshop continues the tradition of addressing challenging problems in a field where the diversity of applications is matched only by the opportunities that await a practitioner.

We would like to thank the authors and the attendees for contributing to the success of this workshop. Special thanks to the referees for reviewing the manuscripts submitted.

Chandrika Kamath (on behalf of the workshop organizers)
Center for Applied Scientific Computing
Lawrence Livermore National Laboratory

SCHEDULE FOR THE FOURTH WORKSHOP ON MINING SCIENTIFIC DATASETS

(PRESENTERS IN **BOLDFACE**)

8:50 – 9:00	Introductory remarks

SESSION 1

9:00 – 9:30	"Clustering Earth Science Data: Goals, Issues and Results", **Michael Steinbach**, Pang-Ning Tan, Vipin Kumar, University of Minnesota, Steven Klooster, Christopher Potter, NASA Ames Research Center, Alicia Torregrosa, California State University, Monterey Bay
9:30 – 10:00	"Flexible Earth Science Data Mining System Architecture", Rahul Ramachandran, Helen Conover, Sara Graves, Ken Keiser, Sunil Movva, and Steve Tanner, University of Alabama in Huntsville
10:00- 10:30	"Clustering of Extra-Tropical Cyclone Trajectories using Mixtures of Regression Models", **Scott Gaffney**, Padhraic Smyth, University of California, Irvine and Andy Robertson, University of California, Los Angeles.
10:30 – 11:00	Break

SESSION II

11:00 – 11:30	"Mining of Topographic Features from Large-Scale Planetary Imagery", **Rie Honda** and Osamu Konishi, Kochi University and Yuichi Iijima, Institute of Space and Astronautical Science, Japan.
11:30 – 12:00	"Support Vector Machines and Kernel Fischer Disriminants: A Case Study using Electronic Nose Data", Dennis DeCoste, Michael Burl, Jet Propulsion Laboratory and Alan Hopkins and Nathan S. Lewis, California Institute of Technology.
12:00 – 1:30	Lunch (on your own)

SESSION III

1:30 – 2:00	"Time-invariant Sequential Association Rules: Discovering Interesting Rules in Critical Care Databases", **Jafar Adibi** and Wei-Min Shen, University of Southern California
2:00 – 2:30	"Modeling Sparse Engine Test Data Using Genetic Programming", **Tina Yu** and Jim Rutherford, Chevron
2:30 – 2:45	Break

SESSION IV

2:45 – 3:15 "Discovering Corrosion Relationships in Eddy Current Non-Destructive Test Data", Donald E. Brown, University of Virginia, Charlottesville and John R. Brence, US Military Academy, West Point

3:15 – 3:45 "Damage Prediction and Estimation in Structural Mechanics Based on Data Mining", S. S. Sandhu, R. Kanapady, K.K. Tamma, and V. Kumar, University of Minnesota, and C. Kamath, Lawrence Livermore National Laboratory

3:45 – 4:15 "Determination of an Initial Mesh Density for Finite Element Computations via Data Mining", R. Kanapady, S. K. Bathina, K. K. Tamma, and V. Kumar, University of Minnesota, and C. Kamath, Lawrence Livermore National Laboratory

Clustering Earth Science Data: Goals, Issues and Results[*]

Michael Steinbach[+] Pang-Ning Tan[+] Vipin Kumar[+]
Steven Klooster[+++] Christopher Potter[++] Alicia Torregrosa[+++]

[+] Department of Computer Science and Engineering, Army HPC Research Center
University of Minnesota
{steinbac, ptan, kumar@cs.umn.edu}

[++] NASA Ames Research Center
{cpotter@mail.arc.nasa.gov}

[+++] California State University, Monterey Bay
{klooster,atorregrosa@gaia.arc.nasa.gov}

ABSTRACT

This paper reports on recent work applying data mining to the task of finding interesting patterns in earth science data derived from global observing satellites, terrestrial observations, and ecosystem models. Patterns are "interesting" if ecosystem scientists can use them to better understand and predict changes in the global carbon cycle and climate system. The initial goal of the work reported here (which is only part of the overall project) is to use clustering to divide the land and ocean areas of the earth into disjoint regions in an automatic, but meaningful, way that enables the direct or indirect discovery of interesting patterns. Finding "meaningful" clusters requires an approach that is aware of various issues related to the spatial and temporal nature of earth science data: the "proper" measure of similarity between time series, removing seasonality from the data to allow detection of non-seasonal patterns, and the presence of spatial and temporal autocorrelation (i.e., measured values that are close in time and space tend to be highly correlated, or similar). While we have techniques to handle some of these spatio-temporal issues (e.g., removing seasonality) and some issues are not a problem (e.g., spatial autocorrelation actually helps our clustering), other issues require more study (e.g., temporal autocorrelation and its effect on time series similarity). Nonetheless, by using the K-means as our clustering algorithm and taking linear correlation as our measure of similarity between time series, we have been able to find some interesting ecosystem patterns, including some that are well known to earth scientists and some that require further investigation.

Keywords

K-means clustering, time series, earth science data, scientific data mining

1. INTRODUCTION

The project team to which we belong is a group of computer and ecosystem scientists focusing on the development of algorithms and tools to help ecologists discover changes in the global carbon cycle and climate system. These techniques will aid ecologists in their efforts to better understand global scale changes in biosphere processes and patterns, and the effects of widespread human activities, such as deforestation, biomass burning, industrialization, and urbanization. Ecologists who work at the regional and global scale have identified Net Primary Production (NPP) as a key variable for understanding the global carbon cycle and the ecological dynamics of the Earth. NPP is the net assimilation of atmospheric carbon dioxide (CO_2) into organic matter by plants. Terrestrial NPP is driven by solar radiation and can be constrained by precipitation and temperature. Keeping track of NPP is important because it includes the food source of humans and all other animals and thus, sudden changes in the NPP of a region can have a direct impact on the regional ecology. An ecosystem model for predicting NPP, CASA (the Carnegie Ames Stanford Approach [PKB99]), has been used for over a decade to produce a detailed view of terrestrial productivity.

Our project uses the multi-year output of CASA, as well as other climate variables, such as long term sea level pressure, sea surface temperature (SST) anomalies, etc., to discover interesting patterns relating changes in NPP to land surface climatology and global

[*] This work was partially supported by NASA grant # NCC 2 1231 and by Army High Performance Computing Research Center contract number DAAH04-95-C-0008. The content of this work does not necessarily reflect the position or policy of the government and no official endorsement should be inferred. Access to computing facilities was provided by AHPCRC and the Minnesota Supercomputing Institute.

climate. Predicting NPP based on, for example, sea surface temperature, would be of great benefit given the near real-time availability of SST data and the ability of climate forecasting to anticipate SST El Nino/La Nina events. For a number of years, ecosystem scientists on our team have used traditional statistical tools for spatio-temporal data analyses relating NPP and other climate variables. Data mining [KH99] can complement these statistical tools in many ways, e.g., some of the steps of hypothesis generation and evaluation can be automated, facilitated and improved.

In this paper we report on a portion of the work involved in this project. In particular, the initial goal of the work reported here is to use clustering to divide areas of the land and ocean into disjoint regions in an automatic, but meaningful way that enables us to identify regions of the earth whose constituent points have similar short-term and long-term climate characteristics. Given relatively uniform clusters we can then identify how various ecosystem phenomena, such as El Nino, influence the climate and NPP of different regions.

There are significant issues related to the spatial and temporal nature of earth science data: the "proper" measure of similarity between time series, the seasonality of the data, and the presence of spatial and temporal autocorrelation (i.e., measured values that are close in time and space tend to be highly correlated, or similar). Although sophisticated approaches to time series similarity are available, e.g., dynamic time warping, we chose standard linear correlation as our similarity measure since it works well with our clustering algorithm (K-means) and lends itself to statistical tests. Since earth science data has a very cyclical (e.g., seasonal) nature, and since earth scientists are mostly interested in non-seasonal patterns, we typically used a couple of preprocessing techniques (moving average and monthly Z-score) to remove seasonality from the data before clustering. However, these seasonality removal techniques affect the degree of temporal autocorrelation of the data (both positively and negatively), and hence, affect the "significance" of the observed correlations. On the other hand, the high degree of spatial autocorrelation of the earth science data we are analyzing actually is beneficial, allowing our K-means clustering algorithm to produce clusters consist mostly of a relatively small number of geographically contiguous regions.

The basic outline of this paper is as follows. Section 2 provides a description of the earth science data. Section 3 describes our clustering technique, which is based on K-means. Section 4 discusses related clustering work and Section 5 considers the issue of how to preprocess the data to remove seasonality patterns. Section 6 describes our initial results in applying clustering to earth science data, while section 7 is a short conclusion and an indication of future directions.

2. Earth Science Data

The earth science data for our analysis consists of global snapshots of measurement values for a number of variables (e.g., NPP, temperature, pressure and precipitation) collected for all land surfaces or water (see Figure 1). These variable values are either observations from different sensors, e.g., precipitation and sea surface temperature (SST), or the result of model predictions, e.g., NPP from the CASA model, and are typically available at monthly intervals that span a range of 10 to 50 years. The attribute data within a global snapshot is represented using spatial frameworks, i.e., a partitioning of the Earth's surface into a set of mutually disjoint regions which collectively cover the entire surface of Earth. For the analysis presented here, we focus on attributes measured on latitude-longitude spherical grids of different resolutions, e.g., NPP, which is available at a resolution of 0.5° x 0.5°, and sea surface temperature, which is available for a 1° x 1° grid.

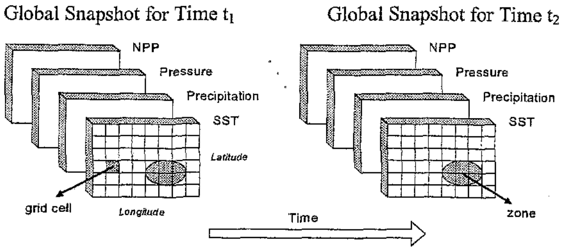


Figure 1: A simplified view of the problem domain.

Using variables derived from sensor observations, earth scientists have developed standard climate indices. These indices are useful because 1) they can distill climate variability at a regional or global scale into a single time series, 2) they are related to well-known climate phenomena such as El Nino, and 3) they are well-accepted by earth scientists. For example, various El Nino related indices, such as ANOM1+2 and ANOM4, have been established to measure sea surface temperature anomalies across different regions of the Pacific Ocean. (El Nino is the anomalous warming of the eastern tropical region of the Pacific, and has been linked to various climate phenomena such as droughts in Australia and heavy rainfall along the western coast of South America.) Some of the well-known climate indices are shown in Table 1 [IND1, IND2]. Figure 2 shows the time series for the ANOM1+2 index. Note that the peak in 1982 and 1983 corresponds to a severe El Nino event.

Climate Index	Description
SOI	Measures the sea level pressure (SLP) anomalies between Darwin and Tahiti
NAO	Normalized SLP differences between Ponta Delgada, Azores and Stykkisholmur, Iceland
ANOM 1+2	Sea surface temperature anomalies in the region bounded by 80°W-90°W and 0°-10°S
ANOM 4	Sea surface temperature anomalies in the region bounded by 150°W-160°W and 5°S-5°N
NP	Area-weighted sea level pressure over the region 30N-65N, 160E-140W

Table 1: Description of well-known climate indices.

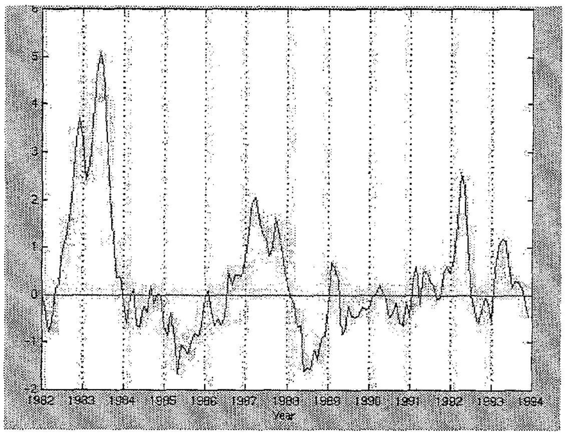


Figure 2: ANOM 1+2 time series.

3. A K-means Based Clustering Approach

Clustering, often better known as spatial zone formation in this context, segments oceans and land into smaller pieces that are relatively homogeneous in some sense. While these zones can be specified directly by researchers, clustering provides a general data mining approach for automatically creating zones. Thus, our basic approach is to treat the zone creation problem as a cluster analysis problem [DJ88, KR90]. Cluster analysis groups objects (grid cells) so that the objects in a group are similar to one another and different from the objects in other groups. The clusters produced may be nested (hierarchical) or un-nested (partitional), overlapping or non-overlapping.

For our initial clustering approach, we chose the widely used K-means clustering algorithm [DJ88], which is simple and efficient. As our results will show, it was effective for our use of clustering during exploratory data analysis.

The K-means algorithm discovers K (non-overlapping) clusters by finding K centroids ("central" points) and then assigning each point to the cluster associated with its nearest centroid. (Note that a cluster centroid is typically the mean or median of the points in its cluster and "nearness" is defined by a distance or similarity function.) Ideally the centroids are chosen to minimize the total "error," where the error for each point is given by a function that measures the discrepancy between a point and its cluster centroid, e.g., the squared distance. Note that a measure of cluster "goodness" is the error contributed by that cluster. For squared error and Euclidean distance, it can be shown [And73] that a gradient descent approach to minimizing the squared error yields the following basic K-means algorithm. (Note that the previous discussion still holds if we use similarities instead of distances, but our optimization problem becomes a maximization problem.)

Basic K-means Algorithm for finding *K* clusters.

1. Select *K* points as the initial centroids.
2. Assign all points to the closest centroid.
3. Recompute the centroid of each cluster.
4. Repeat steps 2 and 3 until the centroids don't change (or change very little).

K-means has a number of variations, depending on the method for selecting the initial centroids, the choice for the measure of similarity, and the way that the centroid is computed. For this work, we followed the common practice of using the mean as the centroid and selecting the initial centroids randomly. For our similarity measure, we chose Pearson's correlation coefficient, which is defined as follows: The correlation coefficient r of two data vectors, $\boldsymbol{x}$ and $\boldsymbol{y}$ is given by

$$r = \frac{\sum_i (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_i (x_i - \bar{x})^2 \sum_i (y_i - \bar{y})^2}},$$

where x_i (y_i) is the value of the i^{th} attribute of $\boldsymbol{x}$ (y), and $\bar{x}$ ($\bar{y}$) is the average value of all attributes of $\boldsymbol{x}$ (y). Correlation has a value between –1 (perfect negative linear correlation) and 1 (perfect positive linear correlation), with a value of 0 indicating no linear correlation.

Since we are using correlation instead of Euclidean distance, there is a question of whether K-means will still "work." However, if the data is standardized by subtracting off the mean and dividing by the standard deviation, then a bit of algebraic manipulation will show that the correlation and the Euclidean distance are monotonically related, as shown in following equation

$r(x^*, y^*) = 1 - \frac{d^2(x^*, y^*)}{2n}$, where x^* and y^* are the standardized vectors of dimension n, and r and d are the correlation and Euclidean distance functions, respectively. Thus, the traditional K-means algorithm will "work" when used with correlation. Furthermore, the measure of cluster goodness that corresponds (at least monotonically) to the traditional squared distance is the sum of the similarity of each point in a cluster to the cluster centroid.

We make a brief comment about our reasons for using correlation. First, correlation is insensitive to changes in scale, and since we want to compare time series of different variable types, e.g., NPP and SST, we need this property. Also, correlation has been well studied by statisticians and thus, confidence intervals and tests for non-zero correlation are readily available. Finally, correlation is widely used as a measure of similarity between time series.

4. Related Work

In this section we discuss other techniques that have recently been used to cluster earth science data. The goal is to indicate possible alternatives to K-means, and to further illustrate some of issues involved in clustering earth science data.

In [SID99], a mixture model approach is used to identify the cluster structure in atmospheric pressure data. (Mixture models assume that the data is generated probabilistically from a mixture of Gaussian distributions and use the data to estimate the parameters of these distributions.) This approach is related to K-means [Mit97], but has two advantages. First, it assigns a "membership" probability to each data point and each cluster. These probabilities provide a measure of the uncertainty in cluster membership. Second, it is sometimes possible to estimate the most appropriate choice for K [SID99]. (It is also possible to estimate the best K for K-means by plotting the overall error or similarity for different values of K and looking for the knee in the plot.)

Another possible approach to clustering, particularly in spatially oriented domains, is to use "region growing." Starting with individual points as clusters, each cluster is grouped with the most similar, physically adjacent cluster, until there is only one cluster. (Sometimes various criteria are applied to prevent clusters from being merged if the resulting cluster is too "poor.") This approach can be viewed as a form hierarchical clustering which has the constraint that clusters can only be merged if the resulting cluster is contiguous, i.e., not split into disconnected sets of points [Mur95].

However, it is sometimes desirable to have clusters that are "piecewise contiguous," i.e., consist of points which are similar, but not all in one contiguous region. An example such an approach is presented in [Til98] and was applied to the problem of land use classification based spectral image data. The technique, Recursive Hierarchical Image Segmentation, consists of alternating steps in which similar, adjacent, regions are merged (a region growing step) and similar, non-adjacent regions are merged (a spectral clustering step). For land use classification, this allows the grouping of points, which may represent the same type of land cover, but which are in disconnected regions. (The K-means approach that we use will automatically produce piecewise contiguous regions.)

Perhaps the work that is most closely related to ours is [Viv00}, which introduces ACTS (Automatic Classification of Time Series), a clustering method for remote sensing time series. (The data considered is NDVI, the Normalized Difference Vegetation Index, or greenness index [NASA].) The goal of this work was to use clustering as an initial step for deriving continental-scale to global-scale vegetation maps. After the removal of components with a period of one year or less, clustering was also used to group points that had similar patterns of inter-annual variation in NDVI. However, there was no investigation of the relationships between different regions of the land and the ocean.

While there has been considerable research into hierarchical clustering and spatial clustering [HKT01], many issues still remain. Some of the new issues of zone formation are zonal formation over time, the multi-scale nature of the data, and constrained zone formation.

5. Dealing with the Seasonality of Data

Another important task in our research work is the removal of seasonal variation from the time-series data. Mostly, earth scientists are interested in non-seasonal patterns, instead of the yearly patterns of (Spring, Summer, Fall, Winter) or (Rainy Season, Dry Season). It is not that these patterns are unimportant, but rather that they are well known, and the events of interest are deviations from the normal seasonal patterns that represent long term cycles, e.g., decadal oscillations, or trends, e.g., global warming. Given such a focus, and the strength of the seasonal patterns in the data, it is necessary to remove them to see other patterns.

There are several ways to do this and Figure 3 shows the results of applying two different types of transformations (filtering) to a particular time series of values. In particular, we focus on a sample time series for sea surface temperature. (This time series was derived from data corresponding to a ½° by ½° region

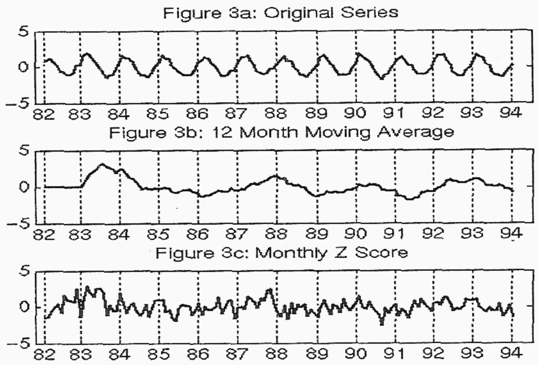


Figure 3: Effects of data pre-processing to remove seasonal variation.

of the ocean at 71.5° W, 23° S, just off the Eastern coast of South America.) This original time series, which clearly has a strong seasonal pattern, is shown by Figure 3a.

While we briefly show the effects of two different types of transformations, these issues and other time series specific issues are discussed in more detail in a related paper [Tan+01]. (Among other issues, that paper discusses the removal of seasonality based the use of DFT (Discrete Fourier Transform and SVD (singular value decomposition.) To allow all the time series to be displayed on a similar scale, all time series were standardized by subtracting off the mean and dividing by the standard deviation.

Moving average. A 12-month moving average is effective in removing seasonality and also smoothes the data significantly. However, as discussed in [Tan+01], a moving average increases the magnitudes of the observed correlations, and at the same time, makes these higher correlations less meaningful. Figure 3b shows the 12-month averaged time series.

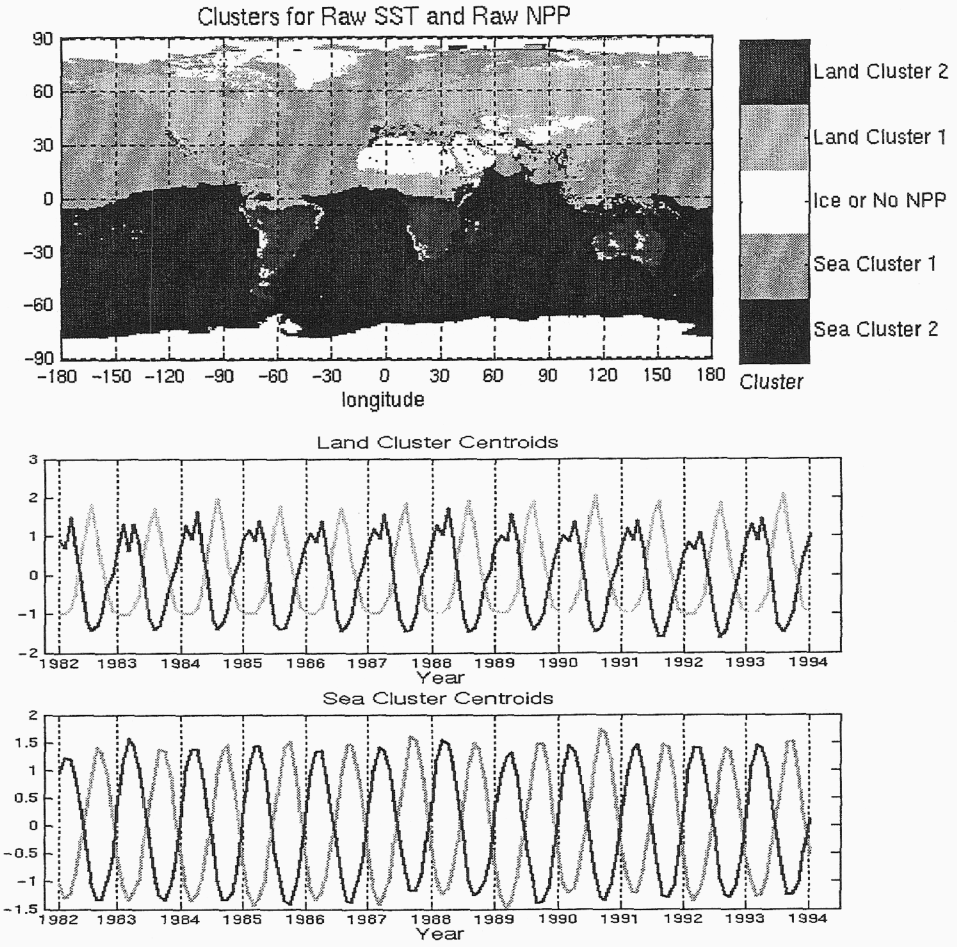


Figure 4. Two Ocean (SST) and Land (NPP) Clusters.

Monthly Z score. This transformation takes the set of values for a given month, calculates the mean and standard deviation of that set of values, and then "standardizes" the data by calculating the Z-score of each value, i.e., by subtracting off the corresponding monthly mean and dividing by the monthly standard deviation. This is slightly different from the usual statistical (Z score) standardization of subtracting the mean and dividing by the standard deviation, since each data point is standardized by using the mean and standard deviation of the values for its month, not the overall mean and standard deviation. Since it removes seasonality (but does not smooth), the monthly Z score transformation reduces autocorrelation [Tan+01]. The result of applying a monthly Z score filter is shown in Figure 3c.

6. Results

In this section we show the use of clustering for detecting different sorts of ecosystem patterns. To do this we employ two kinds of diagrams. The first diagram shows which points on the globe belong to specific clusters by associating each cluster with a particular color. The second type of diagram plots the cluster centroids. Since the cluster centroids are time series, this type of a plot can show various types of temporal patterns. For example, for a cluster consisting of land points, each of which is characterized by a series of monthly NPP values, the centroid of a cluster provides a "summary" description of NPP for the points in that cluster.

Finding Seasonal Patterns and Anomalous Regions. Figure 4 shows the result of finding two clusters for NPP and (separately) finding two clusters for SST. (Note that the seasonal component has not been removed from this data.) The four clusters approximate the northern and southern hemispheres, for land and ocean. The plots of the land and sea centroids show strong yearly cycles. Interestingly, while the northern and southern hemisphere land clusters are mostly contiguous, some areas in the northern hemisphere, e.g., part of southern California, correspond to the "southern hemisphere" cluster and vice-versa. These regions correspond to climates, e.g., a Mediterranean climate, whose plant growth patterns are reversed from those typically observed in the hemisphere in which they reside. The existence of these anomalous climate regions is well known, but clustering allows them to be easily detected.

Identifying Connections between Land and Ocean Clusters. Another use of clustering is to investigate the relationship of various land and sea areas. In particular, by finding land and sea clusters that are highly correlated, we can identify potential teleconnection patterns, i.e., recurring and persistent climate patterns that span vast geographical areas. This works as follows. A large number of clusters are found for the land (NPP) and the sea (SST), say 100 for each. Then the correlations between various sea and land centroids are calculated, and the land and sea clusters with the highest correlations are plotted. Figure 5 shows such a diagram for sea cluster 19 (which is a region of ocean off the coast of Japan) and land clusters 56 (which consists of parts of Japan and

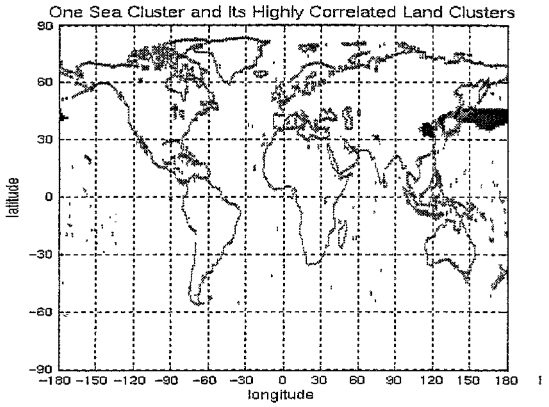


Figure 5: One Sea Cluster and Highly Correlated Land Clusters.

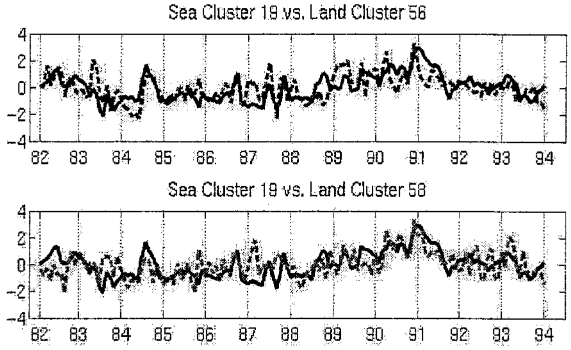


Figure 6: Comparison of Cluster Centroids.

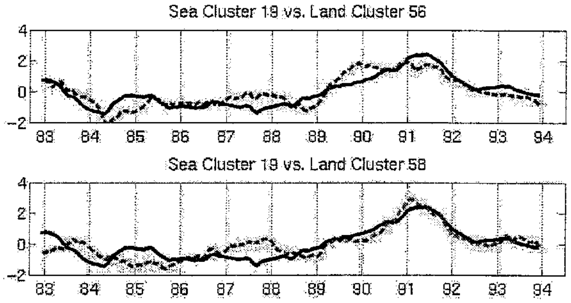


Figure 7: Comparison of Smoothed Cluster Centroids.

Korea, and a region near Pakistan-northwestern India) and 58 (which consists of part of China near the coast). The NPP centroids of land clusters 56 and 58 are correlated with the SST centroid of sea cluster 19 at a level of 0.56 and 0.50, respectively. (For this analysis we removed seasonal variation by using the monthly Z score.) Figures 6 shows a plot of the centroid of sea cluster 19 versus the cluster centroids of land clusters 56 and 58. To better display the overall relationships between the centroids, Figure 7 shows the same centroids after they have been smoothed using a 12-month moving average.

Unlike the pattern that we found in the previous section, the teleconnection pattern displayed in Figure 5 between the sea region (sea cluster 19) and the land regions (land clusters 56 and 58) is not well known to ecosystem scientists. While further investigation by ecosystem scientists is needed to determine whether these relationships are meaningful or not, these clustering results have at least provided the basis for an initial hypothesis. In particular, it would be interesting to see whether the teleconnection between sea cluster 19 and the region near Pakistan-northwestern India can be verified, since these regions are far apart.

Sea cluster 19 is highly correlated (-0.77), with one of the ocean indices, PDO, which is a long-lived El Niño-like pattern of Pacific climate variability. The new hypothesis suggested by this apparent teleconnection is that ENSO (El Nino Southern Oscillation) influences NPP in the Pakistan-India region through variations in seasonal rainfall patterns. This type of El Nino association with rainfall has been noted before for the Indian subcontinent. As the mean sea level pressure difference between the south central Pacific (e.g. Tahiti) and the Indian Ocean weakens, the trade winds can relax, monsoons become weaker, and there can be strong drought in India and Australia. This relationship was noted as far back as 1904 by Sir Gilbert Walker, a British mathematician serving the British Colonial Service. However, the monsoonal teleconnection pattern to ENSO events has not been consistently strong in recent times, (see [KRC99]), which means that more work is required on our part to better understand the patterns shown in Fig 5.

Finding Correlations between Land Clusters and (Ocean) Climate Indices. We also investigated the land-ocean connection by using climate indices that are based on the SST or pressure differences, either between two points on the ocean or over an area of the ocean (see Table 1). For example, some of the indices relate to the El Nino effect. These indices are also time series and thus, we can find the clusters on the land and sea that display a strong correlation to a particular index. Figure 8 shows the land and sea clusters that correlate highly (positive or negative correlation of 0.5 or above) to three different climate indices: PDO (Pacific Decadal oscillation) and two El Nino indices, ANOM 4 and ANOM 1+2 [IND1, IND2]. For this analysis we removed seasonal variation by using the monthly Z score. The ocean regions that are highly correlated with the two El Nino indices are related to the regions used to define the two indices.

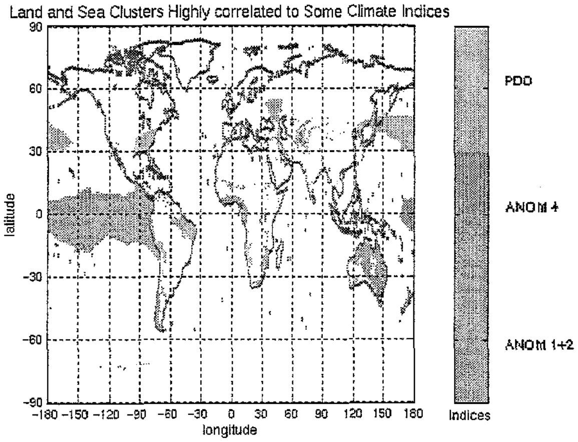


Figure 8: Clusters that are Highly Correlated with Climate Indices

To illustrate the potential for clustering to find interesting teleconnections between land and ocean regions, note that there is a land cluster near Zimbabwe, in southern Africa, which is highly correlated to the ANOM 1+2 index. A connection between southern African rainfall and the El Nino phenomenon has been observed. For instance, Ropelewski and Halpert [RH96] have shown a positive correlation between the southern Oscillation Index (SOI) (another El Nino related climate index) and southern African rainfall. More specifically, the droughts which have occurred in southern Africa since the end of the 1960s are associated with warmer temperatures in the eastern and central tropical Pacific, in the tropical Indian Ocean, and in the equatorial Atlantic. The spatial structure of these anomalies may be associated with El Nino/La Nina events.

7. Conclusions

A key conclusion of this paper is that clustering can play a useful role in the discovery of interesting ecosystem patterns. The patterns revealed by the clusters and their associated (centroids) time series are sometimes well known, e.g., the yearly seasonal variation of Figure 4. However, we have also started to investigate how clustering might be used to discover previously unknown relationships between regions of the land and sea. In this effort, we have focused on climate indices, which are time series of temperature or pressure that correlate well with certain regions of the ocean from which they are derived. In particular, we have looked at which regions of the land are most highly correlated to these centroids. So far the ecologists on our team have found the results interesting and have recognized some familiar patterns. One challenge is to find techniques to automatically select interesting patterns and eliminate spurious ones.

To produce meaningful clusters it is necessary to take into account the spatio-termporal nature of the data. Seasonality must be removed by using appropriate pre-processing steps if non-seasonal patterns are to be detected, and there are significant issues concerning what levels of correlation between time series indicate significant connections. However, on positive side, it is likely that the simple K-means clustering approach we are using works as well as it does because of the high level of spatial auto-correlation in the data. Otherwise, the clusters produced by K-means might consist of a large number of widely separated small regions. The use of clusters that are only piecewise contiguous has not been a problem so far, although much of the evaluation proceeds via visualization and people are good at noticing interesting patterns and ignoring noise. The chief insights come when the clusters consist mostly of large, coherent areas, although, in such cases, the exceptions to the rules can also be interesting as with the case of Figure 4 and southern California.

In clustering, there are a number of opportunities for future research. For instance, we could try other similarity measures, e.g., Euclidean distance or the cosine measure. We could also try the other clustering approaches mentioned in Section 4 or variants of K-means, e.g., bisecting K-means [SKK00]. Along somewhat different line, we may want to look at clusters that vary over time or we may want to try to define clusters in terms of events. (However, for some transformations of the data, e.g., the monthly Z score, we are in some sense already looking at events, i.e., deviations from the norm.) Also, our current clustering approach only looks at the time series for one variable for each point. This is a potential limitation in terms of the goodness of the clusters and their suitability for predicting the behavior of one region (cluster) based on the time varying behavior of another region.

Other limitations in our approach result from the fact that often, only extreme events that are correlated. For example, the El Nino indices have values for each month of each year, but the effects of El Nino on other regions often occur only when the index has an extreme value, i.e., when an El Nino effect is actually occurring. Although there may be a number of possible ways to address these problems and make the clustering more effective, it seems likely that some patterns will best be detected by other data mining techniques that are naturally more event-based, e.g., association rules or co-location rules. Nonetheless, we are hopeful that our clustering approach, and any improvements that we make to it, will continue to produce interesting and useful results.

REFERENCES

[And73] Michael R. Anderberg, *Cluster Analysis for Applications*, Academic Press (1973).

[DJ88] R. C. Dubes and A. K. Jain, *Algorithms for Clustering Data*, Prentice Hall (1988).

[HKT01] J. Han, M. Kamber and K. H. Tung, "Spatial Clustering Methods in Data Mining: A Survey", Harvey J. Miller and Jiawei Han (eds.) (2001), *Geographic Data Mining and Knowledge Discovery,* Taylor and Francis, forthcoming (expected 2001).

[IND1] http://www.cgd.ucar.edu/cas/catalog/climind/

[IND2] http://www.cdc.noaa.gov/USclimate/Correlation/help.html

[KH99] M. Kamber, and J. Han, *Data Mining: Concepts & Techniques*, Morgan Kaufmann (1999).

[KR90] L. Kaufman and P. J. Rousseeuw, *Finding Groups in Data: an Introduction to Cluster Analysis,* John Wiley and Sons (1990).

[KRC99] K. K. Kumar, B. Rajagopalan, and M. A. Cane, "On the weakening relationship between the Indian monsoon and ENSO," *Science*, 284, 2156-2159 (1999).

[Mit97] Tom Mitchell, *Machine Learning*, McGraw Hill (1997).

[Mur95] F. Murtagh, "Contiguity-constrained hierarchical clustering," In I.J. Cox, P. Hansen and B. Julesz, eds., *Partitioning Data Sets,* DIMACS, AMS, 143-152 (1995).

[NASA] http://earthobservatory.nasa.gov/Library/

[PKB99] C.S. Potter, S. A. Klooster, and V. Brooks, "Inter-annual variability in terrestrial net primary production: Exploration of trends and controls on regional to global scales," *Ecosystems*, 2(1): 36-48 (1999).

[RH96] C. F. Ropelewski and M. S. Halpert, "Quantifying Southern Oscillation - precipitation relationships", *J. Climate*, 9,1043-1059 (1996).

[SIG99] Padhraic Smyth, K. Ide, and M. Ghil, "Multiple Regimes in Northern Hemisphere Height Fields via Mixture Model Clustering," *Journal of Atmospheric Science*, 56, 3704-3723 (1999).

[SKK00] Michael Steinbach, George Karypis, and Vipin Kumar, "A Comparison of Document Clustering Techniques," *Text Mining Workshop, KDD 2000.* Boston, MA (2000).

[Tan+01] Pang-Ning Tan, Michael Steinbach, Vipin Kumar, Steven Klooster, Christopher Potter, Alicia Torregrosa, "Finding Spatio-Termporal Patterns in Earth Science Data: Goals, Issues and Results," Submitted to KDD Temporal Data Mining Workshop, KDD2001 (2001).

[Til98] J. C. Tilton, "Image Segmentation by Region Growing and Spectral Clustering with a Natural Convergence Criterion," *Proc. of the 1998 International Geoscience and Remote Sensing Symposium (IGARSS '98)*, Seattle, WA (1998).

[Viv00] N. Vivoy, "Automatic Classification of Time Series (ACTS): a new clustering method for remote sensing time series," *International Journal of Remote Sensing* (2000)

Flexible Earth Science Data Mining System Architecture

Rahul Ramachandran*, Helen Conover, Sara Graves, Ken Keiser, Sunil Movva and Steve Tanner
Information Technology and Systems Center
University of Alabama in Huntsville
(256) 824-5157
rramachandran@itsc.uah.edu

ABSTRACT

The Information Technology and Systems Center (ITSC) at the University of Alabama in Huntsville developed the Algorithm Development and Mining (ADaM) system under a research grant from NASA to investigate new methods of processing large volumes of Earth Observing System (EOS) remote sensing data sets. This system provides knowledge discovery and data mining capabilities for data values as well as for metadata, and catalogs the information discovered. ADaM incorporates algorithms for detecting a variety of geophysical phenomena to address the needs of the earth science community. This data mining system has been used for other research studies dealing with topics such as texture classification, image processing and statistical analysis of earth science data sets. This paper will provide a detailed description of the ADaM system architecture, design, components, client interface and the processing environment. It will also describe the future directions that ITSC intends to pursue with ADaM.

Keywords
Data Mining, ADaM, Content Based Search, ERSS, Mining Plan Builder

1. INTRODUCTION

Designing a data mining system for Earth Science applications is complex and challenging. The issues that need to be addressed in the design are (1) variability of data sets, (2) operations for extracting information, and (3) providing the capability to the user to write complex mining plans. Earth Science data sets not only come in different formats, types and structures; there are also many different states of processing such as raw data, calibrated data, validated data, derived data or interpreted data. The mining system architecture must be designed to be flexible to handle these variations in data sets. The operations required in the mining system vary for different application areas within Earth Science. Operations could range from general-purpose operations such as image processing techniques or statistical analysis to highly specialized, data set-specific science algorithms. The mining system architecture should be flexible in its ability to process new data sets and incorporate new operations without too much effort. The design of the architecture should also allow other users to build new clients to utilize such a system. The Information Technology and Systems Center at the University of Alabama in Huntsville originally developed the Algorithm Development and Mining (ADaM) system under a research grant from NASA Headquarters Research Announcement (NRA) to investigate new methods of processing large volumes of Earth Observing System (EOS) remote sensing data sets. ADaM is designed to handle the complexity of mining Earth Science data [1,2]. It can process heterogeneous data sets and allows users to add research problem-specific science algorithms to the system.

This paper will discuss issues that had to be considered in designing a flexible system architecture. It will describe the ADaM system and its user interface as an example of a flexible design. This paper will also describe the research directions that are evolving from this innovative architecture.

2. DESIGN ISSUES

As stated in the introduction, several issues had to be considered while designing a flexible mining system for Earth Science. These are:

2.1 Data Handling Capabilities

Earth Science data introduces complexity in designing, building and utilizing a data mining system, because these data sets can be quite varied. They can be point data collected by a meteorological instrument, swath or grid data collected by satellites, or volume scan data collected by weather radar. The formats of these data sets also vary from simple binary or ASCII files to more complex structures such as Hierarchical Data Format for the Earth Observing System (HDF-EOS). The spatial and the temporal resolutions of these data sets depend upon the measuring instrument and the platform. The spatial resolution could vary from hundreds of kilometers to a few meters. The temporal range of a data file could vary from 15 minutes to a day or longer. Temporal resolution could vary from instantaneous measurements to accumulation of data over some period. To utilize mining techniques over the broad range of data sets, the mining system had to be designed to handle these types of data set variations.

2.2 Addition of New Algorithms

In certain circumstances, a known scientific algorithm can be utilized to extract the information needed from data sets. Detecting Mesoscale Convective Systems (MCS) from SSM/I data utilizing the Devlin [3] algorithm is one such example. The data mining system had to be designed to be flexible enough to allow not only data set specific algorithms but also other new algorithms to be added to it without affecting the other operations.

2.3 Allow Scientists to Select and Sequence Different Operations

The Mining system also needed the capability to allow scientists to create their own mining plans. A mining plan is a sequence of specified steps, where each step is a processing operation. The scientist should be able to piece together different operations/algorithms to reach their goal.

3. ADaM SYSTEM FEATURES

The ADaM system was designed using the latest object Oriented techniques to achieve a high degree of portability, accessibility and modularity. The implementation in standard C++ allows the system to run on multiple operating systems including IRIX, Linux, and Microsoft Windows NT. One of the design goals was to have ADaM work at both data archive centers or on a user's desktop workstation.

3.1 Overview of the Architecture

The ADaM data mining system has been designed to extract content based metadata from large Earth Science data archives. It can detect phenomena or events that are of interest to scientists and then store this information in a way that facilitates the data search and order process. Some mining results are stored in Event/Relationship Search System (E/RSS), an ITSC-developed spatial data search engine used to find coincidences between mining-generated phenomena, climatological events and static information such as country and river basin boundaries [4]. The data mining engine also provides other data ordering related capabilities such as subsetting and custom data product generation through specialized client applications. Custom processing may include gridding, resampling, filtering, format conversion, or other analysis depending on the needs of the customers. For example, ADaM can generate a monthly total rain accumulation image from radar reflectivity data. Both the E/RSS and custom processing client are web applications, so the clients are capable of running in almost any environment. Figure 1 depicts a generalized view of how the ADaM data mining architecture has been utilized. This architecture allows the clients to communicate to the system in a variety of ways such as: (1) The miner engine can be driven directly via local scripts or an interactive console session, (2) A web application can guide the user in creating mining plans, which execute the mining engine, (3) A network application can submit mining plans via the miner daemon, and (4) The system may also be used as a library with the application directly linking to the individual operations needed.

3.2 Processing Flow

The ADaM system architecture is based on a processing stream, in that mining is broken down into a series of steps with results from each step passed to the next one in line. Figure 2 illustrates both ADaM's data processing stream, as well as the three basic types of modules: input, processing, and output. The use of data input filters, specialized for a variety of data types, has been instrumental in simplifying the development of the

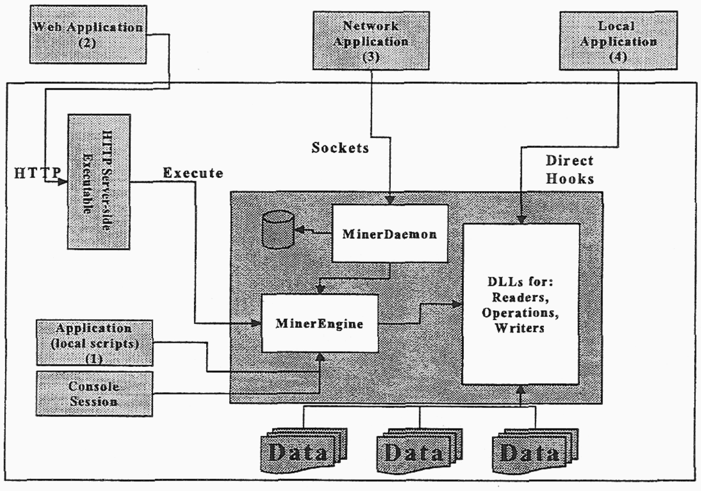


Figure 1: Multiple Process Flows Utilizing the ADaM Data Mining Architecture

processing and output operations. The selected input filter translates the data into a common internal structure so that the processing operations can all be written for a single data representation.

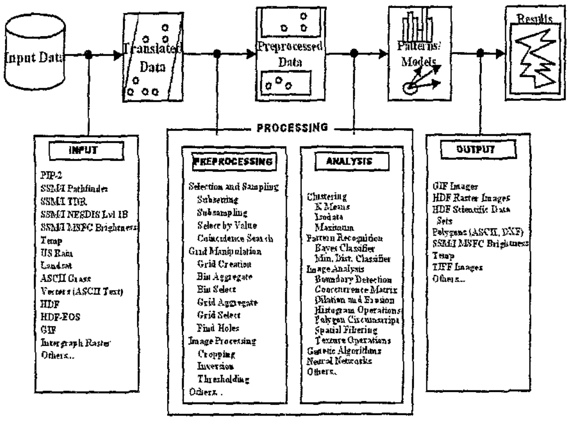


Figure 2: Schematic diagram depicting the stream of a mining system

This allows the addition of new operations to the system without having to address input data format problems. Similarly, the addition of a new input filter provides access to the entire suite of processing operations for the data type in question. This design feature allows ADaM to handle heterogeneous Earth Science data sets. The mining system currently allows over 120 different operations to be performed on the input data stream. These operations vary from specialized atmospheric science data set specific algorithms to generalized image processing techniques. The last step in the mining process is the selection of the input modules, the output filters effectively insulate the processing operations from having to support all the possible output formats.

output format. Since the input data has been converted to ADaM's internal format, the output modules allow the user the option to select either the input format or a different format for the final data product. In the same manner as

3.3 Components

In order to allow for the distributed use of the data mining functionality, the ADaM system was designed as a client-server architecture, which supports remote client applications communicating with the data mining server. This allows the server system to be co-located with archived data stores while being driven by either remote or local clients. In support of this architecture, the ADaM data mining system is composed of the mining engine and mining daemon, both located on the server. The daemon supports a specific protocol of messages and listens on a configured port for instructions from client applications.

Through instructions from the client, the daemon is responsible for managing user access information, file management operations and job scheduling and management. The daemon ultimately sends the correct information and directions to the engine in the form of a "mining plan" for actual processing. A software interface layer was created providing tools to assist client application developers in communicating with the mining daemon across network sockets. Figure 3 depicts the connections between the components of the ADaM client-server architecture. Each component performs a specific well-defined task, and therefore the components themselves may be replaced or updated provided that the new components conform to the same interfaces.

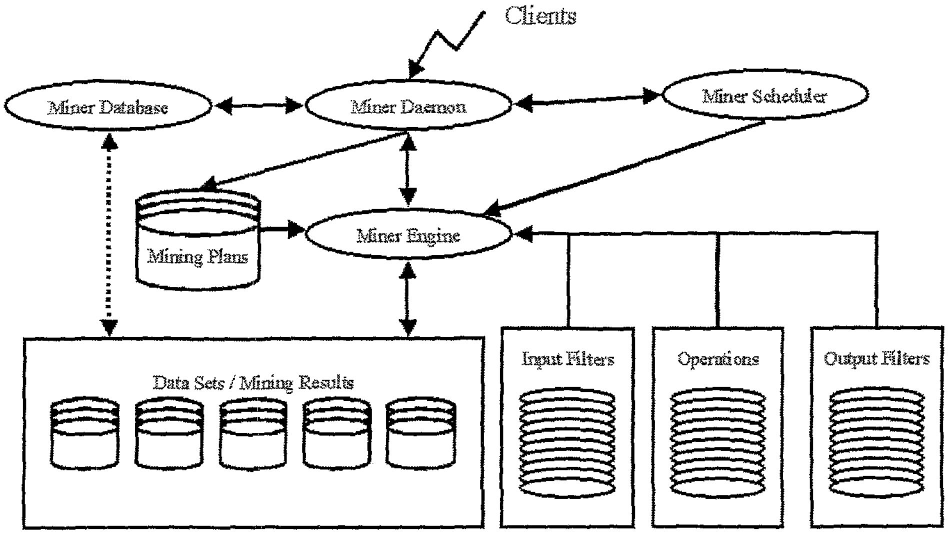


Figure 3: AdaM Data Mining Server Components

3.3.1 Mining Engine:

The Mining Engine is the software component that manages the processing of data through a series of specified operations. The input, processing and output modules are dynamically loaded as needed at execution time, and this allows for the addition of newly developed modules without the need to rebuild the engine. The mining engine interprets a mining plan script that provides the details about each specified operation and the order that they should be executed. Other communication with the mining engine is managed through the mining daemon process.

3.3.2 Mining Daemon:

The Mining Daemon is the gateway to the mining engine. All network communications with the mining system are handled by the daemon through a message handling protocol. Upon installation the daemon is configured to listen on a specific port for any socket communications. The daemon is capable of handling a fairly rich set of messages that allows it to perform file management duties, command the mining engine and provide user security screening. The daemon can also determine at run time which processing modules are available on the server.

3.3.3 Mining Database:

The database component is used to store information that is required for the smooth operation of the system and the interaction of its components. This information includes the names, locations and related metadata for input data sets available on the server. It also includes information about users, jobs, mining results, and other related information. A relational database is currently used for this task. Access to the database is provided by the daemon.

3.3.4 Mining Scheduler:

The scheduler component examines the list of jobs to be executed on the server and determines which job or jobs to execute at any given time. The scheduling policy used can be unique to each server. The scheduler invokes the mining engine for each job and monitors its progress, updating the job status in the database whenever it changes.

3.3.5 Operations and Data set Input/Output Filters:

Each of the operations and data set filters is implemented as a shared library. The libraries are loaded dynamically by the mining engine, which means that new modules may be added to the system without recompiling or relinking. Each of the operations and filters is completely independent of all the others. All operations and filters either produce or operate on a common format representing scientific data. This design feature allows science specific algorithms to be incorporated into the system with relative ease.

3.3.6 Mining Plans :

The mining plan script conveys the processing instructions to the mining engine. The plan contains the number and sequence of processing steps as well as the detailed parameters (tokens) describing how to perform each step, such as where to find the input data, where to store the output and configuration parameters for all the various operations. Mining plans may be created using the mining plan editor. Since mining plans are text files, they may also be created using any text editor. It is easy for applications to write mining plans. The mining plan begins with a number indicating the number of operations in the plan. The remainder of the plan is a series of token/value pairs where the tokens and values are delimited by newlines.

4. ADaM PLAN BUILDER CLIENT

In order to allow users to build complex mining plans, ITSC has designed an easy to use and functional user interface called the ADaM Plan Builder. This user interface is a client that communicates directly to the mining engine. It makes it easier for the user to select the right operation for the task and to provide values for the parameters for that operation. The individual ADaM operation documentation is written in XML^{TM}[5]. Since this standardizes the documentation, the Plan Builder written in Java parses and utilizes the information contained in those XML files. Thus, the Plan Builder Interface utilizes these XML files to provide the user options on the operations available, what parameters each operation requires, the meaning of each parameter, default values for those parameters and finally a sample mining plan. Through the Plan Builder, users can select sample operation steps and modify the values for the parameters according to their needs. The ADaM Plan Builder allows the user to chain together complex mining plans for scientific research.

The Plan Builder also allows the users to edit and modify the Mining Database. The user can feed the metadata information about the data files to be mined into the database via this client. The database then automatically selects the correct files for mining based on the time range given. The architecture of the Plan Builder is shown in Fig. 4 and a screen capture of the interface is shown in Fig. 5.

5. RESEARCH DIRECTIONS

ADaM is currently undergoing a metamorphosis in the sense of becoming uncoupled from an environment that is dependent on centralized processing on a single server platform with the availability of local data. The following sections describe some of the efforts underway to migrate ADaM into a highly distributed environment that will provide broader access to the system and distributed heterogeneous scientific data sets, while addressing improved scalability and flexibility.

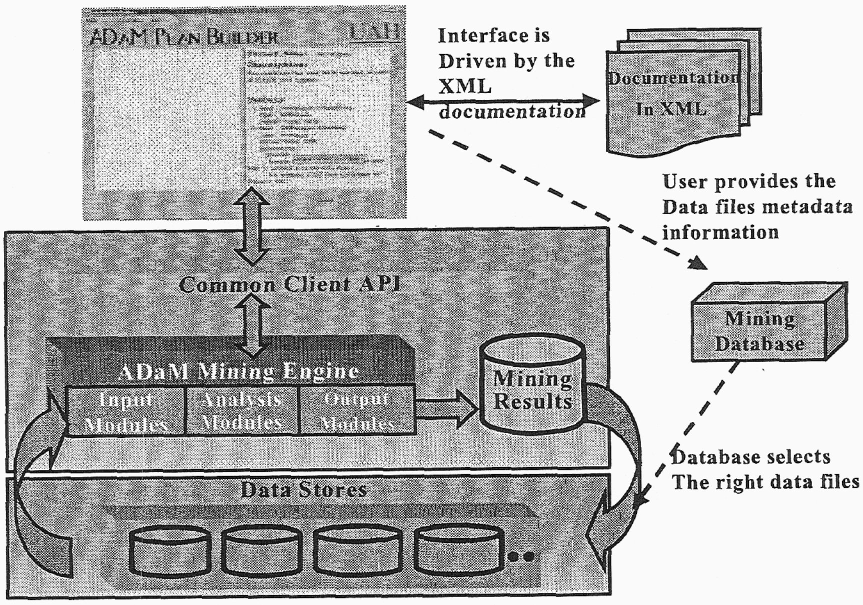


Figure 4: ADaM Plan Builder Functional Diagram

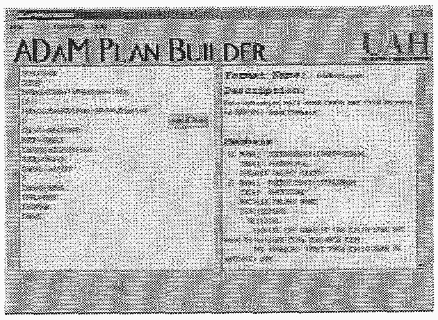


Figure 5: Screen capture of the ADaM Plan Builder Interface

5.1 Distributed Mining

ITSC is currently investigating and prototyping emerging distributed component technologies. To address the use of distributed mining services and access to distributed data sets. The use of distributed mining services opens the system to greater possibilities of extensibility, performance, scalability and reliability by distributing the processing burden and lessening the possibility of centralized points of system failure [6]. Current research efforts have been successful with the development of an Earth Science Markup Language [http://esml.itsc.uah.edu] that will make great strides towards realizing generic access to heterogeneous data sets [7]. The integration of ESML technology with planned distributed mining components is expected to result in a virtual processing environment that capitalizes on improved networking bandwidth and under-utilized distributed processors.

5.2 Grid Mining

Another approach to distributed mining is also being prototyped in the form of Grid Mining. ITSC researchers, in collaboration with NASA/Ames researchers, have been successful with implementing and testing the ADaM system on the NASA Information Power Grid [8]. The Grid approach employs a sophisticated infrastructure of message passing, scheduling and security in an effort to utilize large capacity processing and data centers for scientific research. This approach to distributed data mining promises to be of particular benefit to scientific researchers in need of massive processing and data resources.

5.3 Mining Onboard Space Craft

ITSC is also investigating and developing an innovative processing system capable of handling the unique constraints and characteristics of the on-board satellite data and information environment. The EnVironmEnt for On-Board Processing (EVE) system will serve as a proof-of-concept of advanced information systems technology for remote sensing platforms [9]. EVE's on-board, real-time processing will provide capabilities focused on the areas of autonomous data mining, classification and feature extraction. These will contribute to Earth Science research applications, including natural hazard detection and prediction, fusion of multi-sensor measurements, intelligent sensor control, and the generation of customized data products for direct distribution to users. EVE is being engineered to provide high performance data processing in a real-time operational environment. A ground-based testbed is being created to provide testing of EVE and associated Earth Science applications in a heterogeneous embedded hardware and software environment.

6. CONCLUSION

ADaM has proven to be an effective and valuable tool to mine Earth Science spatial data [10,11]. Its flexible architecture design has made it possible for ADaM to handle the multiple formats, scales, resolutions and large granule sizes typical of spatial data for many different science problems. The design permits the easy addition of new algorithms, especially domain-specific science algorithms. The ability of the user to create complex mining plans by chaining together different operations is also possible because of the flexibility of the architecture. ITSC plans to utilize its experience in designing ADaM to meet the scientific mining requirements of the next generation of scientists in several other domains. Research efforts are focused towards distributed mining across the web; mining large volumes of data on the Information Power Grid; and finally designing a system that would be used onboard aircraft or spacecraft to extract features or phenomena as soon as they are sensed by the instrument.

7. REFERENCES

[1] Hinke, T.H., J. Rushing, H. Ranganath and S. J. Graves, "Techniques and Experience in Mining Remotely Sensed Satellite Data," Artificial Intelligence Review (AIRE, S4): Issues on the Application of Data Mining, pp 503-531, 2001.

[2] Keiser, K., J. Rushing, H. Conover, and S. J. Graves, "Data Mining System Toolkit for Earth Science Data," Earth Observation (EO) & Geo-Spatial (GEO) Web and Internet Workshop, Washington, D.C., February 1999.

[3] Devlin, K, 1995: Application of the 85 GHz ice scattering signature to a global study of mesoscale convective systems. Master's thesis, Texas A&M University, August 1995.

[4] Hinke, T. H., J. Rushing, S. Kansal, S. J. Graves, H. Ranganath, E. Criswell, "Eureka Phenomena Discovery and Phenomena Mining System," AMS 13th Int'l Conference on Interactive Information and Processing Systems (IIPS) for Meteorology, Oceanography and Hydrology, 1997.

[5] XML: Extensible Markup Language, http://www.w3.org/XML

[6] Fu, Yongjian, "Distributed Data Mining: An Overview", 8th IEEE International Conference on Network Protocols, November 2000.

[7] Ramachandran, R., M. Alshayeb, B. Beaumont, H. Conover, S. J. Graves, N. Hanish, X. Li, S. Movva, A. McDowell, and M. Smith, "Earth Science Markup Language," 17th Conference on Interactive Information and Processing Systems for Meteorology, Oceanography, and Hydrology, 81st American Meteorological Society (AMS) Annual Meeting, Albuquerque, NM, January, 2001.

[8] Hinke, Thomas, J. Novotny, "Data Mining on NASA's Information Power Grid", Proceedings of the Ninth IEEE International Symposium on High Performance Distributed Computing, Pittsburgh, Pennsylvania, August 1-4, 2000

[9] Steve Tanner, Ken Keiser, Helen Conover, Danny Hardin, Sara Graves, Kathryn Regner, and Matt Smith, EVE: An Environment for On-Orbit Data Mining, IJCAI Workshop on Knowledge Discovery from Distributed, Dynamic, Heterogeneous, Autonomous Data and Knowledge Sources, Seattle, Washington, August 4-10, 2001

[10] Ramachandran, R., H. Conover, S. J. Graves, K. Keiser, "Algorithm Development and Mining (ADaM) System for Earth Science Applications," Second Conference on Artificial Intelligence, 80th AMS Annual Meeting, Long Beach, CA, January, 2000.

[11] Rushing, J., H. Ranganath, T. Hinke, S. Graves, "Using Association Rules as Texture Features", IEEE Transactions on Pattern Analysis and Machine Intelligence, 2001.

Clustering of Extra-Tropical Cyclone Trajectories using Mixtures of Regression Models

Scott Gaffney
Department of Information and Computer Science
University of California, Irvine
Irvine, CA 92697-3425
sgaffney@ics.uci.edu

Andy Robertson
Department of Atmospheric Sciences
UCLA
Los Angeles, CA 90095-1565
andy@atmos.ucla.edu

Padhraic Smyth
Department of Information and Computer Science
University of California, Irvine
Irvine, CA 92697-3425
smyth@ics.uci.edu

ABSTRACT

Extra-tropical cyclones (ETCs) are responsible for severe and highly damaging winter weather over North America and western Europe; they cause the second largest insurance loss due to weather, after tropical cyclones. On the positive side, they are also the primary source of water for much of the western United States. Historical observational data, as well as simulated general circulation model data, provide atmospheric scientists with the opportunity to better understand and predict the dynamics of ETCs and their interaction with other local and global meteorological processes. In this paper we describe our recent work on using mixtures of regression models as a general framework for clustering spatio-temporal trajectory data, and specifically in this context, the clustering of ETC trajectories. The steps necessary to analyze such data are described, including preprocessing, detection and tracking of cyclones, and (finally) clustering. We discuss the end-to-end process of data analysis in this context as well as preliminary results on obtaining ETC clusters and their interpretation from a scientific perspective.

1. INTRODUCTION

With the increasing availability of massive observational and experimental data sets (across a wide variety of scientific disciplines) there is an increasing need to provide scientists with efficient computational tools to explore such data in a systematic manner. For example, in astronomy, techniques such as classification and clustering are widely and successfully used to organize stellar objects into groups and catalogs—which in turn provide the impetus for further scientific hypothesis formation and discovery [1], [2], [3].

Data-driven exploration of massive *spatio-temporal* data sets is an area where there is particular need of such computational tools. Scientists are overwhelmed by the vast quantities of data which simulations, experiments, and observational instruments can produce. For spatio-temporal data, investigators are typically not primarily interested in raw grid-level data, but rather in higher-level phenomena and emergent processes which drive the data such as the temporal and spatial evolution of specific localized structures of interest. Examples include trajectories of birth-death processes for vortices and interfaces in fluid-flow experiments, extra-tropical cyclones (ETCs) from sea-level pressure data over the Atlantic and Pacific oceans, and sunspot shape and size evolution over time from hourly images of the solar photosphere. The ability to automatically detect, cluster, and catalog such objects can in principle provide an important "data reduction front-end" to convert four-dimensional data sets (one temporal and three spatial dimensions) on a massive grid to a much more abstract representation of local structures and their evolution. In turn, these higher-level representations can provide a general framework and basis for further scientific hypothesis generation and investigation, such as investigating correlations between local phenomena and global trends (e.g., how storm paths are related to hemispheric geopotential regime patterns).

In this paper we describe preliminary results on clustering of objects into K groups based on observed spatio-temporal *trajectories* with application to clustering of extratropical cyclones (ETCs). Existing automated tools for clustering and classification have been largely based on the so-called *feature-vector representation* of an object. Concatenating measurements such as object brightness, shape, and size yields a vector which can be viewed as existing in a Euclidean space. Useful notions such as distance, similarity, decision boundaries, prototypes, and clusters then spring forth from the natural geometry of this space.

Feature-vector methods, however, have significant limitations when applied directly to trajectories (e.g., by representing a sequence of T position measurements (x_t, y_t) as a vector of length $2T$). For example:

- By lumping all measured attributes together the vector representation loses information about locality in space and time. (For example, smoothness of object trajectories is lost.)
- Different objects have trajectories of varying lengths

T and may evolve at different time-scales and involve various birth-death mechanisms.

- Feature-based models are unable to model systematically the relationship of the object's spatial evolution to features such as size, shape, and velocity.

2. MODEL-BASED CLUSTERING OF TRAJECTORIES USING FINITE MIXTURES

We have developed a general framework for clustering trajectories using probabilistic mixture models of dynamic systems which can systematically handle the above issues in a principled manner. In [4] we described a general probabilistic framework for clustering individual objects given observed trajectory data. The key idea is to use a generative probabilistic model for each trajectory, and to address the trajectory clustering problem by hypothesizing that each observed trajectory is being generated by one of K such models. Formally, we have a finite mixture of K components, where each component model describes a particular class of trajectories. It is straightforward to show that, given the functional form of the trajectory models, one can use the Expectation-Maximization algorithm to infer the parameters of the K models given a set of N trajectories [5].

In addition one can estimate posterior probability of membership in each group for each object [6], [7]. This provides a coherent and sound mechanism for clustering objects based on their observed dynamic behavior, and provides a principled and straightforward framework for issues such as handling trajectories of different lengths, coupling of input or covariate information (e.g., [8]), etc. Furthermore, the probabilistic formalism can be used to objectively guide the selection of the best model to explain the data, enabling search over different underlying dynamic model structures as well as search for the best value of K (e.g., [2], [9], [8]).

3. SCIENTIFIC BACKGROUND ON EXTRATROPICAL CYCLONES (ETCS)

The primary application of trajectory clustering that we have investigated up to this point is the clustering of ETC tracks from meteorological data. ETCs are important for a number of reasons. They are responsible for severe and highly damaging winter weather over North America and western Europe. They are also the primary source of water for much of the western United States.

Atmospheric scientists are interested in the spatio-temporal patterns of evolution of ETCs for a number of reasons. It is not well-understood how long-term climate changes (such as global warming) may influence ETC frequency, strength, and spatial distribution. Given the significant impact of ETCs (both from a potential damage viewpoint, and as mechanisms for water supplies), there is significant motivation to be able to understand the potential correlation of climate change with patterns of ETC occurrence. Similarly, changes in ETC patterns may provide clues of long-term changes in the climatic processes that drive ETCs. The links between ETCs and local weather phenomena are also of interest: clearly ETCs have significant influence on local precipitation, and in this context a better understanding of their dynamics could provide better forecasting techniques both on local and seasonal time-scales. Furthermore, an explicit model of ETC evolution can serve as an intermediate link between global atmospheric phenomena (e.g., geopotential height patterns and regimes) and local "weather-related" phenomena such as precipitation, addressing the long-standing problem in atmospheric science known as "downscaling" (i.e., how to link and model global-scale and local-scale phenomena in a coherent manner).

4. PRIOR WORK ON CLUSTERING ETC TRAJECTORIES

A useful starting point for modeling ETCs is to try to identify different subclasses of ETCs based on their observed trajectories. The work of Blender et al in [10] is illustrative of the use of conventional clustering techniques in atmospheric science for clustering in this context. Using sea-level pressure data on a grid over the North Atlantic (measurements every 6 hours, available over several winters) Blender et al detect local minima in the pressure map and then use a nearest-neighbor tracking algorithm (forward in time, with some spatial distance constraints) to connect up the minima in successive maps and determine trajectories. The trajectories are then converted into a fixed-dimensional vector for clustering by the k-means algorithm: Blender et al require their storm trajectories to be exactly 3 days in length, resulting in 12 (x, y) pairs which are converted to a 24-dimensional vector, one per trajectory. Based on subjective analysis of the data, $k = 3$ clusters are chosen and fit in this 24-dimensional space. Despite the somewhat ad hoc nature of the approach the resulting clusters demonstrate that storms in the North Atlantic clearly cluster into different types of trajectory paths. Blender et al use the resulting clusters to then classify each day into the cluster of the dominant trajectory path for that day and analyze the regional average pressure maps (or "regimes") for each set of days.

5. EXPERIMENTS WITH MIXTURE-BASED CLUSTERING OF ETCS

To begin the examination of the utility of our model-based clustering techniques with ETC trajectories, we used several simulation datasets and analyzed them using a somewhat simple scheme based on our general framework. The datasets that we are working with are part of the CCM3 AMIP II simulation data model runs. Specifically, we have data for the winter months (November to April) from 1980 to 1985 that give mean sea-level pressure (MSLP) measurements on a $2.5° \times 2.5°$ grid over the earth every 6hrs. Of course, since we are interested in ETCs over the North Atlantic we focused on the area between 30°N-70°N and 80°W-10°E. As an extra step, the data was preprocessed to filter out the long term effects in the measured field.

Our primary interst is in the clustering of cyclone trajectories. However, we must first discover, or track, the cyclones from the raw MSLP data before we can proceed with this step.

5.1 Finding the Minima

Following the lead of Blender et al in [10], we begin our cyclone tracking scheme with a minima finding process. An important aspect of the approach of employing dynamic models for cyclone tracking is that we must be able to track

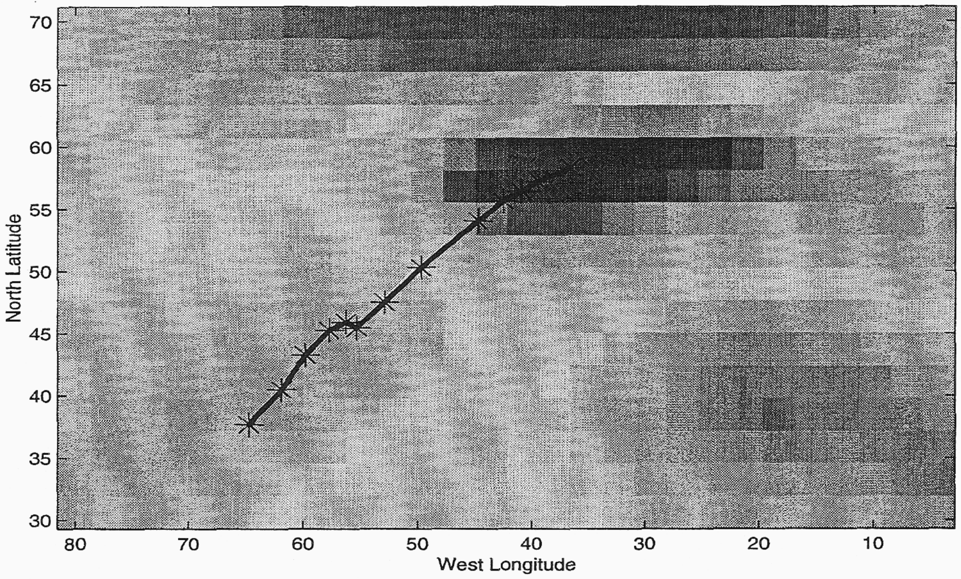


Figure 1: A single generated cyclone trajectory. The line in the figure connects the '*' symbols along the trajectory that represent the minima that were tracked as part of this cyclone. The background image represents the MSLP data at the time the cyclone is centered at the far right '*'.

cyclones in continuous state-space. Of course this means that we cannot limit the possible locations of tracked cyclones to the grid upon which the MSLP data is measured. As such we investigated the inclusion of an interpolation scheme into the minima finding process so that we could afford the tracking of cyclones in continuous state-space.

We use a bicubic interpolation method coupled to an iterative scheme to find minima using simple gradient descent. First we scan all of the frames (the MSLP data slices) over time and find all of the local minima using a simple sliding neighborhood method. We declare a "pixel" to be at a local minimum if its value is less than all eight of its neighbors. Then we use a simple gradient descent with bicubic interpolation to descend to the point "inside" of the pixel that is at an approximate minimum. This point then gives us our approximate off-grid center of a candidate cyclone.

5.2 Tracking the Cyclones

We then take the candidate cyclone centers from above and complete the following two steps to complete the tracking.

First we scan the frames sequentially from beginning to end and look at each candidate cyclone center to attempt to associate it with a candidate cyclone center from the previous frame. If there exists a center within some small neighborhood region in the previous frame surrounding a center in the current frame, then we link them. If there does not exist an associate in the previous frame, then we designate the candidate center in the current frame to have been newly "born."

In the second step, we take the set of associated centers over time and eliminate all those that exist for less than three days—this removes many small, noisy tracks that correspond to local, small-scale weather disturbances not usually considered to be cyclones. The remaining set of associated centers are taken to be actual cyclone trajectories or tracks.

Figure 1 shows an example of a single trajectory that was generated from the above steps. The image shown displays the MSLP data at the instant in time when the cyclone is at the far right of its trajectory.

6. CLUSTERING TRAJECTORIES

An obvious way that one might go about clustering trajectory data is to take all of the n_j measurements for an individual and form a vector $\mathbf{y}_j$ of dimension n_j. Assume for the moment that each individual has the same number of measurements (i.e., $n_j = n$ for all individuals j) and these measurements were all taken at exactly the same x values. We can then treat the set of $\mathbf{y}_j$ trajectories as a set of n-dimensional vectors in an n-dimensional space and use any of a variety of the many clustering methods which operate in vector-spaces.

While this may be a reasonable approach in some applications, it will not always be applicable or appropriate. For many data sets, the trajectories will be of different lengths and may be measured at different time points. In addition, the $\mathbf{y}$ measurements may be multidimensional (e.g., 3d position estimates in tracking the dynamics of a moving object), in which case there is no natural vector representation.

Perhaps more fundamentally, if one converts the data to a vector representation there is a fundamental loss of information about the data, i.e., if we believe from the underlying physics of the data-generating process that the $\mathbf{y}$'s are a smooth function of the x's, then this smoothness information is lost when we convert a sequence of n numbers to an n-dimensional vector of numbers. Thus, intuitively, retaining the notion of trajectory smoothness in our clustering procedure, should generate better data models compared to throwing away this information.

Thus, we employ a *model-based* clustering of trajectories, where each cluster is modeled as a prototype function with some variability around that prototype. A distinct feature of this model-based approach to clustering is the fact that it produces a descriptive interpretable model for each cluster. Since we are estimating smooth functions from noisy data it is natural to use a probabilistic framework.

7. PRELIMINARY RESULTS ON CLUSTERING CYCLONE TRAJECTORIES

In this section we describe some preliminary clustering experiments using linear regression mixture components in our mixture (see the Appendix) with the cyclone tracks that were obtained from the techniques detailed in section 5.2. On average, there were about 50 cyclones tracked each winter over the North Atlantic, some lasting 3 days, others lasting as long as 6. For the sake of simplicity here, we only show results for the 1981-82 winter and we set the number of components in the mixture model to 3. Of course, as we stated earlier, we could learn the optimal number of components in the mixture in a principled manner if we so desired.

In Figure 2 we see the results of this clustering. The top graph shows all of the tracks from the 1981-82 winter that were discovered. The remaining three graphs in the figure show the three resulting clusters. We can see that there appears to be one cluster of cyclones moving rapidly north-to-northeast (third from top), another moving more slowly northeast (second from top), and another one moving in an east-to-northeast direction (bottom).

One nice benefit of our probabilistic framework is that we can easily and naturally build extensions into our model such as the addition of a background, catch-all, cluster that "soaks up" tracks that appear not to belong to any other particular group. In Figure 3 we see some preliminary results when this type of background cluster is added to the framework. As before, three groups are fit to the data, but in addition a fourth background cluster that has fixed parameters is added to the mix. This fourth cluster has mean corresponding to the population mean curve and large variance: it is intended to "attract" particularly noisy trajectories that are not well fit by any other components. The mixing weight for this cluster is, however, estimated from the data during the application of the EM procedure. The resulting background cluster is shown in the bottom graph of Figure 3, and the three remaining clusters are shown in the other three graphs. It appears from this figure that the algorithm was able to return tighter clusters by placing suspect tracks into the background.

8. ONGOING WORK

We are investigating a general framework for clustering trajectories using probabilistic models of dynamic systems which allows one to overcome limitations of simpler feature-vector methods. As a first step we looked at mixtures of simple regression models and applied these ideas to the problem of clustering ETCs. With the clustering of ETCs, one must

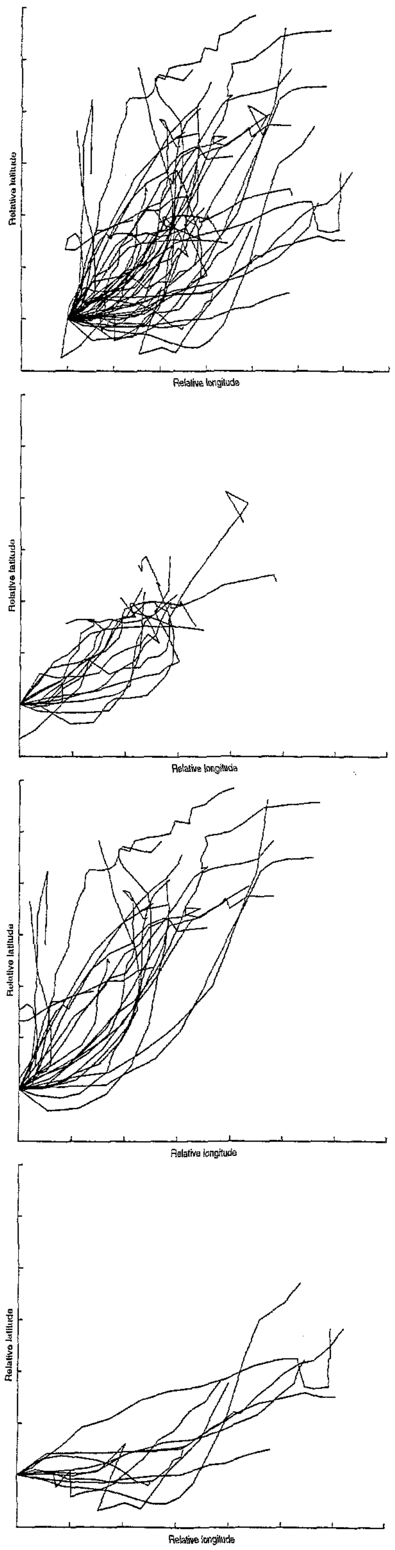


Figure 2: Preliminary results of cyclone clustering using 3 linear regression components. The top graph shows all of the tracks that were provided to the clustering algorithm, the other three graphs show the three resulting clusters.

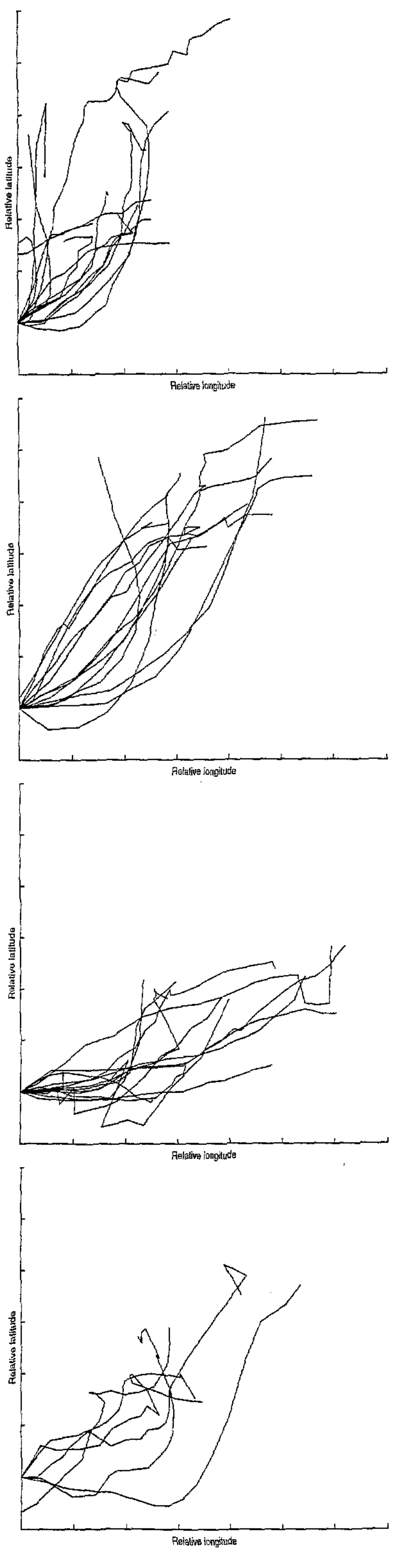


Figure 3: Preliminary results of cyclone clustering using 4 linear regression components with a single background cluster. The bottom graph shows the background cluster, while the other three show the three remaining clusters.

also solve the initial problem of tracking the cyclones themselves before the clustering can begin. It is apparent that the tracking itself can be very noisy, and one can easily dedicate a large amount of time perfecting such a technique. Of course, this is not our intention since our primary goal is to investigate clustering algorithms for objects whose tracks are already known. Nonetheless, one can identify many circumstances, such as merging trajectories, where there is considerable ambiguity in the tracking process. Since these trajectories are the basis for the clustering work, if the tracks themselves are tainted (e.g., noisy), then the clustering will be unstable or non-informative.

A potentially useful general idea is to integrate the tracking with the clustering. In this framework the tracking will drive the clustering, but also the clustering will drive the tracking, i.e., both tracking and clustering will be coupled and estimated jointly, which is inherently more optimal compared to carrying out each estimation separately.

Other future work can focus on several different aspects of the problem. For example, one important aspect is the ability to integrate better component models into the process that allow for the modeling of dynamic behavior, such as AR models or Kalman filters. This will allow the modeller more power in the clustering of trajectories into their respective groups. Furthermore, extensions like the integration of shape and velocity information into the clustering process might also provide some benefits. In preliminary work to date we have found that the AR and Kalman filter models appear to be significantly less stable (from a parameter estimation) viewpoint) than the simpler regression models; thus, it may be that the resolution of the ETC trajectory data is not sufficient to allow accurate modeling of this nature.

Lastly, the evaluation of these techniques against other baseline methods is also important (e.g., comparison with K-means or Gaussian mixtures). Although these evaluations would be worthy of investigation, they are non-trivial to carry out since vector-based methods (such as k-means) don't directly handle variable length trajectories measured at different times. However, one can resort to truncation and/or other techniques to deal with this problem.

Appendix: Mathematical Background

We can define a probabilistic cluster model for sets of trajectories as follows. Let our data set S consist of n_j measurements for each of M individuals, $1 \leq j \leq M$. We will refer to these measurements as being a function of time (i.e. x is synonymous with time), although this is not strictly necessary. Let the trajectory of measurements for the jth individual be denoted as y_j, with the ith measurement of y_j denoted as $y_j(i)$. Furthermore, suppose that the trajectory of measurements y_j were taken at the times in x_j. Finally, let each trajectory in S belongs to one of K groups.

The probability of observing a particular measurement $y_j(i)$, given $x_j(i)$ and component model k, is defined as $f_k(y_j(i)|x_j(i), \theta_k)$, and is assumed to be a conditional regression model. We can then define the probability of a complete trajectory, given a particular component model k

as

$$\begin{aligned} P(y_j|x_j,\theta_k) &= P(y_j(1),\dots,y_j(n_j)|x_j(1),\dots,x_j(n_j),\theta_k) \\ &= \prod_i^{n_j} f_k(y_j(i)|x_j(i),\theta_k). \end{aligned} \quad (1)$$

Here we make the standard regression assumption that, conditioned on the model and the x values (and, thus, the means for the y's are known), the noise is independent at different x points along the trajectory. Dependent noise could be modeled if appropriate for a particular application.

When we don't know which component generated that trajectory (as is the case in practice for clustering), the conditional density of the observed data $P(y_j|x_j)$ is a mixture density:

$$P(y_j|x_j,\theta) = \sum_k^K f_k(y_j|x_j,\theta_k)w_k, \quad (2)$$

where $f_k(y_j|x_j,\theta_k)$ are the mixture components, w_k are the mixing weights, and θ_k is the set of parameters for component k.

Conditional independence between trajectories, given the model, amounts to assuming that our individuals constitute a random sample from a population of individuals, and allows the full joint density to be written as:

$$P(Y|X,\theta) = \prod_j^M \sum_k^K w_k \prod_i^{n_j} f_k(y_j(i)|x_j(i),\theta_k). \quad (3)$$

The log-likelihood of the parameters θ given the data set $\mathcal{S}$ can be defined directly from Eq. (3).

$$\mathcal{L}(\theta|\mathcal{S}) = \sum_j^M \log \sum_k^K w_k \prod_i^{n_j} f_k(y_j(i)|x_j(i),\theta_k). \quad (4)$$

The task at hand is to pull the mixture components out of the joint density, using $\mathcal{S}$ as a guide, so that the underlying group behavior can be discovered. The problem would be simple if it were known to which group each trajectory belonged. Given the group membership of each trajectory, and assuming some particular form for the density functions f_k (e.g., linear regression models with Gaussian noise), the K models can simply be fit to the grouped data. If, however, the group memberships are hidden, as is the case in practice, more complex procedures are required.

A common approach for dealing with hidden data is to employ the EM algorithm ([11], [12]). Gaffney and Smyth in [5] detail extensively the solution for the EM algorithm with these models. It turns out that the solution simply requires using weighted least squares to perform the necessary calculations.

9. ACKNOWLEDGEMENTS

This work was supported by a grant from the Institute for Scientific Computer Research at Lawrence Livermore National Laboratory and by the National Science Foundation under NSF Grant IRI-9703120.

10. REFERENCES

[1] Fayyad U.M., Djorgovski S.G., and Weir N. Automating the analysis and cataloging of sky surveys, In *Advances in Knowledge Discovery and Data Mining*, U. M. Fayyad, G. Piatetsky-Shapiro, P. Smyth, and R. Uthurusamy (eds.), Menlo Park, California: AAAI Press. 471-493, 1996.

[2] Cheeseman, P. and Stutz. J. Bayesian classification (AutoClass): theory and results. In *Advances in Knowledge Discovery and Data Mining*, U. M. Fayyad, G. Piatetsky-Shapiro, P. Smyth, R. Uthurusamy (eds.), Cambridge, MA: AAAI/MIT Press, pp. 153–180, 1996.

[3] Fayyad, U. M. and Smyth, P. Cataloging and mining massive databases for science data analysis, *Journal of Graphics and Computational Statistics*, 8(3), 589–610, 1999.

[4] Cadez, I. V., Gaffney, S., Smyth, P. A general framework for clustering individuals. In *Proceedings of the ACM 2000 Conference on Knowledge Discovery and Data Mining*, Ramakrishnan, R. and Stolfo, S. (eds.), New York, NY: ACM, 140–9, August 2000.

[5] Gaffney, S. and Smyth P. Trajectory clustering with mixtures of regression models. In *Proceedings of the ACM 1999 Conference on Knowledge Discovery and Data Mining*, S. Chaudhuri and D. Madigan (eds.), New York, NY: ACM, 63–72, August 1999.

[6] Titterington D.M., Smith A.F.M., and Makov U.E. *Statistical Analysis of Finite Mixture Distributions.* New York: Wiley, 1985.

[7] Banfield J.D. and Raftery A.E. Model-based Gaussian and non-Gaussian clustering. *Biometrics*, **49**, 803-821, 1993.

[8] Smyth, P. Probabilistic model-based clustering of multivariate and sequential Data. In *Proceedings of the Seventh International Workshop on AI and Statistics*, D. Heckerman and J. Whittaker (eds.), Los Gatos, CA: Morgan Kaufmann, 299–304, 1999.

[9] Fraley, C. and Raftery, A. E. How many clusters? Which clustering method? Answers via Model-based cluster analysis. *Computer Journal*, 41, 578–588, 1998.

[10] Blender, R., Fraedrich, K., and Lunkeit, F. Identification of cyclone-track regimes in the North Atlantic. *Quart J. Royal Meteor. Soc.*, 123, 727–741, 1997.

[11] Dempster, A.P., Laird, N.M., and Rubin, D.B. Maximum likelihood from incomplete data via the EM algorithm, *J. Royal Stat. Soc. B*, **39**(1), 1-38, 1977.

[12] McLachlan G.J. and Krishnan, T. *The EM Algorithm and Extensions*, New York: John Wiley and Sons, 1997.

Mining of Topographic Features from Large Scale Planetary Imagery

Rie Honda
Kochi University
Akebono-cyo 2-5-1
Kochi, JAPAN 780-8520
honda@is.kochi-u.ac.jp

Yuichi Iijima
Institute of Space and
Astronautical Science
Yoshino-dai 3-1-1,
Sagamihara
Kanagawa, JAPAN 229-8510
iijima@planeta.sci.isas.ac.jp

Osamu Konishi
Kochi University
Akebono-cyo 2-5-1
Kochi, JAPAN 780-8520
konishi@is.kochi-u.ac.jp

ABSTRACT

In this study, a crater detection system for a large-scale image database is proposed. The original images are grouped according to spatial frequency patterns and both optimized parameter sets and noise reduction techniques used to identify candidate craters. False candidates are excluded using a self-organizing map (SOM) approach. The results show that despite the fact that a accurate classification is achievable using the proposed technique, future improvements in detection process of the system are needed.

1. INTRODUCTION

Recent advances in sensors and telemetry systems have increased the amount and quality of imagery available for researchers in fields such as astronomy, earth observation, and planetary exploration. However such advances have also increased the need for a large-scale database of scientific imagery and associated data mining techniques. [1][2][4][9][14][13].

Smyth et al.[13] and Burl et al. [1] developed a trainable software system that learns to recognize Venusian volcanos in a large set of synthetic aperture radar imagery taken by the spacecraft Magellan. A machine leaning approach was adopted because it is easier for geologists to identify feature examples rather than describe feature constraints. Experimental results showed that the system was able to successfully identify volcanoes in similar imagery but performance deteriorated when significantly different scenes were used. Burl et al. also proposed an automated feature detection system for planetary imagery named Diamond Eye[2] which was applied to crater detection and showed a good performance, however, a difficulty similar with the previous study was expected.

Hamada et al.[6] reported on the automated construction of image processing techniques based on misclassification rate and an expert system composed of a large set of image processing modules.

In this paper, attention is focused on two difficulties in feature detection in optically observed image databases. The first is heterogeneity of image quality due to differences in illumination and surface conditions that affect the parameters included in the detection process. The second is the wide range of target feature sizes. For example, the diameter of lunar craters ranges from 1000 km to just 100 m (approximately equal to the size of several pixels in the object space).

In this paper, a feature detection system for a large database of scientific imagery is proposed particularly focusing on detecting features with a wide range of sizes from large scale imagery of various quality at the best performance. The technique is applied to the detection of craters in lunar optical imagery.

2. SYSTEM OVERVIEW

Craters are hollow features of varying size and shape and are frequently observed on solid planetary surfaces. Most craters were formed as a result of meteoroid impact. Their number and size distributions provide significant information about meteoroid activity in the past, the age and rheological properties of the planetary surface. Crater analysis has relied on human visual interpretation because of the difficulties in implementing efficient and accurate automation techniques.

In optical imagery, craters are generally recognized by shadows around the rim and represented according to the illumination conditions. Furthermore, image quality varies due to albedo, surface roughness, and illumination conditions, which further complicates the detection process.

Considering these difficulties, the following detection process is proposed: edge detection filtering, binarization, and circular pattern detection using Hough transforms or a genetic algorithm (GA). Concentrating on edge patterns reduces difficulties caused by changing illumination conditions. However, additional parameters such as the binarization threshold are introduced into the detection process and optimiza-

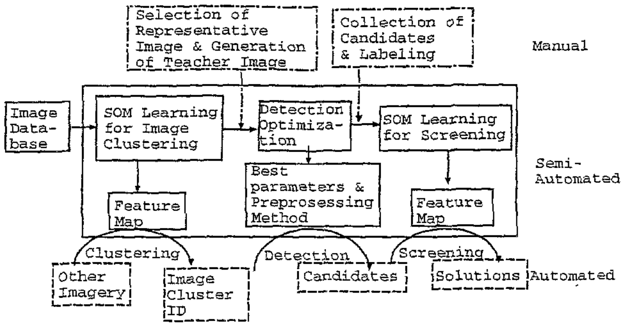

Selection of Representative Image & Generation of Teacher Image
Collection of Candidates & Labeling
Manual
Image Data-base
SOM Learning for Image Clustering
Detection Optimiza-tion
SOM Learning for Screening
Semi-Automated
Feature Map
Best parameters & Preprosessing Method
Feature Map
Clustering
Detection
Screening
Other Imagery
Image Cluster ID
Candidates
Solutions
Automated

Figure 1: System overview.

tion of these parameters should be considered.

Thus the proposed crater detection system is descried as follows.

1. Clustering of original images.
2. Selection of representative image for each cluster and generation of teacher images by manually extracting features.
3. Optimization of detection process for each representative image by comparison with the result of 2.
4. Learning of candidate pattern for solution screening.
5. Detection of feature candidates and screening of unknown images using information obtained in 1 - 4.
6. Storage of extracted feature information in secondary database.
7. High level spatial pattern mining.

A schematic overview of processes 1 to 5 is shown is Figure 1. In this study, processes 1, 2, 3 and 4 are examined in detail and the effectiveness of integrated process evaluated by application to new imagery.

3. CANDIDATE DETECTION

3.1 Crater Detection Method

In this section, the use of Hough transforms and genetic algorithm is shown as possible crater detection modules. Further details of these techniques are provided by Honda et al.[8].

3.1.1 Combinational Hough Transform

Hough transforms are used for the extraction of geometrically simple parametric figures from binary images[10]. For crater detection, the target parameters are the center and the radius of the crater rim. Firstly, the parameter space is divided into cells (bins). Probable parameter values (or trace) are calculated for each signal (white pixel) in a binary image assuming that the signal is a part of the figure, and the count of the corresponding cell is increased by one. After all signals are counted in the parameter space, parameter sets of the figures that exist in the binary image are obtained by extracting parameter cells whose count number exceeds a threshold.

Watanabe and Shibata [15] proposed combinational Hough Transform (CHT) that uses a pair of signals in a restricted region and multiresolution images to simplify projection into a parameter space. The results showed the use of a CHT reduced computation time and significantly improved the solution accuracy. Therefore, a CHT with additional noise reduction and other minor processes to improve accuracy is proposed for crater detection[8].

The algorithm of crater detection based on CHT are summarized as follows.

1. The original binary image are preprocessed by using some of the following methods: isolated noise reduction, expansion and shrinking, thinning by Hilditch's algorithm, pyramid-like signal reduction.
2. The image is degraded using the $W \times W$ pixel filter matrix.
3. The image is divided into the $L \times L$ pixels blocks.
4. The radius of the target circle is set to be $r = L/4$. The following process from 5 to 8 are proceeded increasing r by 1 while $r \leq L/2$.
5. The processes of 6 and 7 are performed for all blocks.
6. Among pairs of white pixels in the block extended by 50% ($2L \times 2L$ pixels), $P_{i1}(x1, y1)$ and $P_{i2}(x2, y2)$, the pairs that satisfy $r \leq |P_{i1}P_{i2}| < 2r$ are selected as signal candidates.
7. The center of the circle (x_{ci}, y_{ci}) is calculated for each pair assuming they exist on a circle rim with radius of r.
8. The couunt of the (x_{ci}, y_{ci}, r) cell in the parameter space is increased by 1.
9. The cells are sorted concerned with number of count. If the count is larger than 0, a circle of (x_{ci}, y_{ci}, r) is projected on the image, and the normalized count and the matching ratio are calculated. The definition of both values are given by
$$NP = p/npp^2, \qquad (1)$$
$$M = bp/npp, \qquad (2)$$
where NP is the normalized count, p is the count, npp is the number of pixels on the rim of projected circle, M is the matching ratio, bp is the number of white pixels of the rim of projected circle. Furthermore, to exclude the false solutions caused by random noises, the internal noise ratio IM within the circle with the radius of hr is introduced, where $0 < h < 1$ (typically $h = 0.6$).
10. The cells satisfying $NP > NP_{threshold} \cap M > M_{threshold} \cap IM < IM_{threshold}$ are extracted as the solutions.

Since the radius of circle is restricted by L, we utilize the multiresolution image of the original grayscale image to detect the circle with the radius larger than $L/2$. It should be noted that appropriate three threshold values and noise reduction methods must be chosen to optimize the performance.

3.1.2 Genetic Algorithm

Genetic algorithms are frequently used to obtain a single solution in optimization problems[5]. In order to implement such an algorithm for circular object detection, based on [12], a gene is set as a binary string that sequentially expresses a parameter set of (x_i, y_i, r_i), where (x_i, y_i) and r_i are the center and radius of the circle represented by the i-th gene, respectively. The fitness of the i-th gene, g_i, is calculated by projecting the circle represented by the i-th gene onto the binary image and checking its overlapping ratio, $g_i = n_i/N_i$, where n_i is the number of white pixels on the circle and N_i is the total number of pixels on the circle.

In order to avoid random noise being incorporated into the solution, we modified g_i as follows:

$$g_i' = g_i - g_{i,r=fr_i}, \tag{3}$$

where $g_{i,r=fr_i}$ is the ratio of the white pixels on a cilcle with a radius of fr_i and $0 < f < 1.0$ (typically $f = 0.3$).

A variety of genes are then randomly produced and evolved through selection, crossing, and mutation. After iteration, genes with a fitness higher than the threshold are extracted as solutions.

Since it is possible to have many solutions (craters) in a single image, a process to unify similar genes and delete detected circles from the original images is introduced to improve the system's ability to detect multiple solutions[8]. After removal of solution circles, genes are newly generated and the process is iterated.

The algorithm of crater detection by GA is summarized as follows.

1. The original image is degraded using $W \times W$ pixel filter matrix.
2. Initial populations of genes are generated.
3. The following process from 4 to 6 are iterated for a given number of generations.
4. The fitness of genes, g_i', are calculated.
5. The genes are selected, crossed, and mutated.
6. The genes with the same attributes are unified.
7. The genes with $g_i' > g_{threshold}'$ are detected as solutions.
8. The solutions are projected as circle rims on the image. The intensity of pixels on the projected circle rims are changed to 0 (black).
9. The processes from 2 to 8 are iterated for a given number of times.

The proposed algorithm also includes several parameters that affect solution accuracy and optimization of these parameters is dependent on image quality.

3.2 Optimization of the Detection Process

Preliminary results of tests using the above method have indicate that noise and signal gaps have a significant deterioration effect on detection accuracy[8]. The optimization of parameters such as the binarization or count threshold is effective technique for improving accuracy, however, the optimized values are dependent on image quality.

The following optimization process is suggested: (1) cluster source images, (2) select representative image from each group, (3) produce teacher image by manual visual recognition, (4) optimize crater detection process by comparing results from teacher images and the result from the corresponding original image. The details of these sub-processes are provided in the following section.

3.2.1 Clustering of Frame Images

It is suggested that the rough grouping of images with respect to image quality is an effective technique for simplifying optimization of the detection process. In this study, the clustering of original images using Kohonen's self-organizing maps (SOM) [11] was examined.

SOM is an unsupervised learning algorithm that uses a two-layer network of input layer and competition layers, both of which are composed of units with n-th dimensional vectors. SOM effectively maps the similar pattern of the input layer on the competitive layer. In the SOM algorithm, the distance (usually Euclidean) between the input vector and each unit vector of the competition layer is calculated and the input vector is placed into the winner unit, which has the smallest distance. At the same time, the unit vectors in the cells adjacent to the winner cell (defined by the neighborhood distance) are modified so that they move closer to the input vector. As a result of this iterative projection and learning, the competitive layer learns to reflect variation of the input vectors and can obtain adequate clustering of the input vectors. Presently, SOM is widely used for the clustering, visualization and abstraction of unknown data sets.

Selection and preprocessing of input vectors is crucial to improve SOM accuracy. In order to group lunar images according to roughness or contrast, the FFT power spectrum of normalized images is adopted as the input vector.

After clustering, a representative image, which has the largest similarity with the unit vector and also includes many craters, is selected for each unit cell (cluster). Then craters are marked in each representative image and binary teacher images generated (see Figure 2(a) and 2(b)).

3.2.2 Optimization of the Detection Process

The detection process is divided into three parts for optimization purposes: binarization, preprocessing including noise reduction, and circle detection. These processes are optimized sequentially using teacher images.

Firstly, edge detection filtering is carried out on the original images. Then, optimal binarization threshold that produce a binary image most similar to the teacher image is identified for each cluster. Based on [6], the evaluation function is defined by

$$\begin{aligned} E_k = \quad & P(T_k(i,j) = S_k(i,j) \mid T_k(i,j) = 1) \\ & -\alpha P(T_k(i,j) \neq S_k(i,j) \mid T_k(i,j) = 0), \end{aligned} \tag{4}$$

where k is the cluster ID, $T_k(i,j)$ is the intensity of the (i,j) pixel of teacher binary image, α is a weight parameter (typically $\alpha = 0.3$), and $S_k(i,j)$ is the intensity of (i,j) pixel of the final binary image defined by

$$S_k(i,j) = \begin{cases} 1 & \text{for } Q_k(i,j) > Q_{th,k} \\ 0 & \text{for } Q_k(i,j) \leq Q_{th,k}, \end{cases} \tag{5}$$

where $Q_k(i,j)$ is image intensity after edge detection filtering and $Q_{th,k}$ is the binarization threshold. The value of $Q_{th,k}$ is searched greedily to maximize E_k.

Next the combination of preprocessing methods that maximizes positive detection rate of craters defined by $Pr_k = N_k/Nt_k$ is identified, where N_k and Nt_k are the numbers of craters detected from the binary image and the teacher binary image for cluster k, respectively.

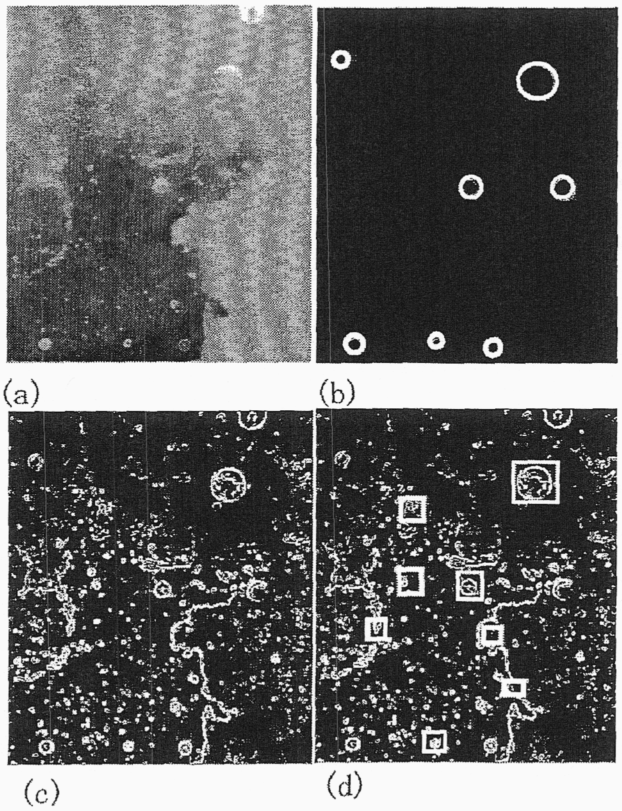


Figure 2: Schematic view of optimization process. (a), (b), (c), and (d) show the original image, teacher image, tuned binary image, and results of detection, respectively. White squares in (d) indicate the extracted candidates.

Finally, the circle detection parameters that maximize Pr_k for the preprocessed image using selected methods is identified. Figure 2 shows a schematic view of the optimization process.

As shown in Figure 2(d), extracted solutions may include many false solutions, which will be excluded in the post-processing stage described in the following section.

3.3 Screening of Solutions

A solution screening process is used in the post-processing stage to exclude false solutions. Candidate crater images are cut out, normalized with respect to its size and intensity, and visually labeled true or false. The candidate pattern is learned by SOM taking the normalized intensity vectors or FFT power spectrum as the input vectors. Each unit in the competition layer is labeled either true or false by evaluating the ratio of candidates in it. If we assume that the properties of the new data set are similar to those of the studied data set, the class (true or false) of the newly detected candidate is decided by projecting it onto the SOM feature map.

4. EXPERIMENTS

4.1 Description of Data Set

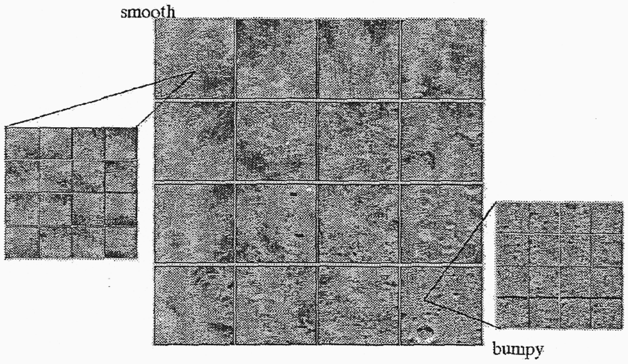


Figure 3: SOM feature map for clustering of original images. Cluster ID 0, 1, 2, . . . , and 15 are for each cell from the upper left corner to the lower right corner in raster-scan order.

A total of 984 medium browse images from Lunar Digital Image Model (LDIM), which had been mosaicked by the U. S. Geological Survey based on the lunar global images obtained by the U. S. Clementine spacecraft. The images were between 322 and 510 pixels in width and 480 pixels in height, and resampled at a space resolution of approximately 500 m/pixel using a sinusoidal projection. Images in the polar regions were not used to avoid distortions due to the map projection. The radius of target craters ranged from 9 to 18 pixels.

For clustering of original images, an area of 256 × 256 pixels was extracted from the center of the normalized images, and the FFT power spectrum calculated as the input vectors of the SOM. The size of the SOM competition layer was defined as 4 × 4 units because only a rough grouping was needed. One hundred images were sampled and the SOM leaning process iterated 100000 times. All images were then projected onto the competition layer, and the unit cell vectors adjusted using K-means method[3]. No images with extremely large distance were identified in this process.

4.2 Result of Image Clustering

Figure 3 shows the resulting competition layer, hereafter denoted the feature map. In this map, the image with the smallest distance with each unit vector is displayed in each cell to visualize the clustering result. It can be seen that relatively smooth images including the Mare recognized by a dark region, are clustered on the left side, and the rugged terrains called the Highland with many clearly identifiable craters are clustered in the lower right corner. This result indicates that learning by SOM successfully distinguishes between variations in image quality and groups them effectively. Based on the clustering result, a representative image was manually selected and a teacher binary image produced for each cluster.

4.3 Result of Detection Optimization

The binarization threshold ranged from 30 to 125 and binary images that approximated the teacher images were produced automatically. It is suggested that a single threshold value for the entire image will not be adequate in some cases be-

Table 1: List of noise reduction cases and optimization result.

ID	thinning	pyramid-like reduction	isolated noise reduction	expansion and shrinking	number of clusters with the best performance
1	No	No	No	No	0
2	No	No	Yes	No	0
3	No	No	No	Yes	1
4	No	No	Yes	Yes	0
5	No	Yes	No	No	0
6	No	Yes	Yes	No	1
7	No	Yes	No	Yes	0
8	No	Yes	Yes	Yes	0
9	Yes	No	No	No	0
10	Yes	No	Yes	No	12
11	Yes	No	No	Yes	0
12	Yes	No	Yes	Yes	0
13	Yes	Yes	No	No	1
14	Yes	Yes	Yes	No	0
15	Yes	Yes	No	Yes	0
16	Yes	Yes	Yes	Yes	0

cause of spatial variations within the image, and that this problem should be solved at the pre-processing stage.

For the optimization of the noise reduction processes, 12 combinations of four noise reduction methods was examined: thinning by Hilditch's algorithm[7], pyramid-like signal reduction, isolated noise reduction, and expansion and shrinking.

Table 1 shows the definition of each noise reduction case and the number of clusters which had the best performance for each case. It can be seen that case 10, which was a combination of thinning and isolated noise reduction, achieved the highest positive solution detection rate for most clusters (12 in 16). Thus a combination of thinning and isolated noise reduction was applied to all clusters in the following process for simplicity.

Figure 4 summarizes the results of the optimization of the detection process for both CHT and GA techniques as performance curves represented by the positive detection rate as functions of the false solution number. In general, a decrease in threshold leads to an increase in both the positive detection rate and the number of false solutions. Figure 4 shows that positive detection rate increases with number of false solutions when the number of false solutions is relatively small, however, it remains constant for lager numbers. Thus the initial point of the flat portion of the performance curve is considered to be the optimum performance condition. In most cases, this coincides with the point that minimizes the false number and maximizes the positive rate.

Figure 4 also shows that CHT performs significantly better than GA. This is mainly caused by the fact that GA's are used to obtain a single solution. Although the GA was modified to obtain multiple solutions, the results show that the GA can acquire only a few solutions per trial even for teacher binary images that include clearly identifiable circles, and requires many iterations to acquire multiple solutions. Thus, CHT was selected as the circle detection module suitable for crater detection.

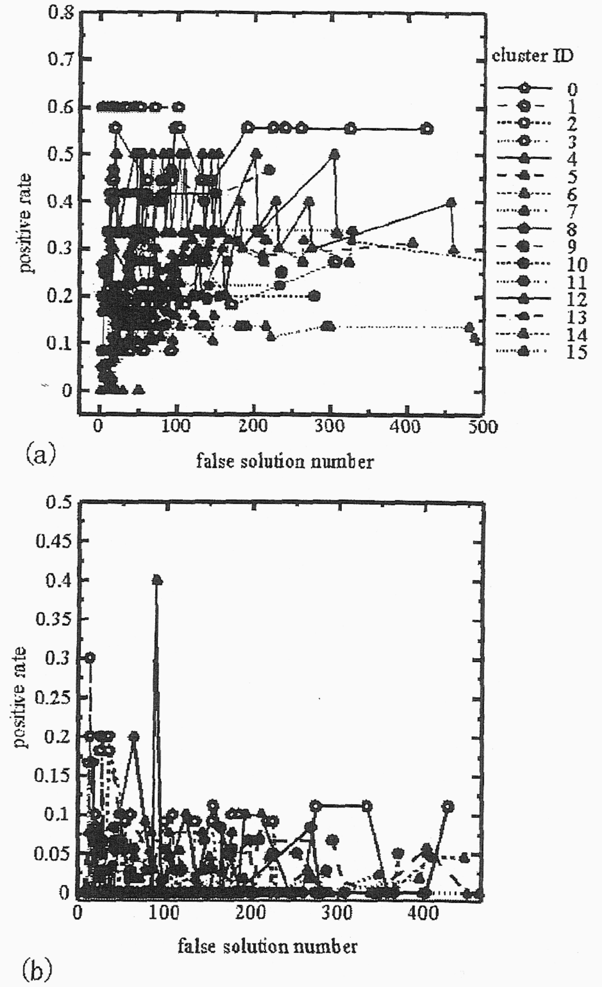


Figure 4: Optimization result for CHT(a) and GA(b).

After optimization, the CHT positive solution detection rate increased significantly up to values ranging from 0.2 to 0.6. However, since information was lost during the binarization process, it will be necessary to consider other detection methods for some clusters to further improve detection performance.

4.4 Result of Screening of Detected Candidates

SOM clustering of crater candidates extracted in the previous process was performed for screening purposes to produce the candidate classifier. A total of 646 candidates were visually labeled either true or false. Half of the candidates were randomly sampled for learning and the remainder were used for examination purpose. The percentage of true candidates for both groups was 25.4% and 27.2%, respectively. All images were rotated such that the direction of sunlight

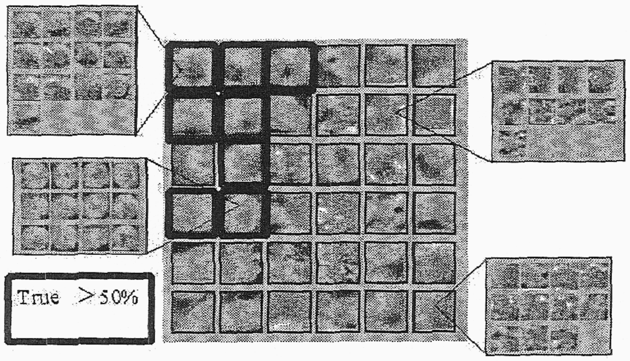


Figure 5: Example of SOM feature map for classification of crater candidates. Input vectors are set to be image vectors.

Table 2: Result of SOM learning for crater candidate screening.

Case	True positive rate	False positive rate	Average positive rate
Image Vector			
study	0.812±0.037	0.931±0.009	0.897±0.016
best/study	0.855	0.946	0.929
test	0.776	0.891	0.864
FFT power spectrum			
study	0.691±0.037	0.803±0.096	0.784±0.013
best/study	0.733	0.788	0.783
test	0.605	0.768	0.755

incidence was equal, and normalized with respect to intensity and size. Two types of input vectors were examined: image vectors represented by pixel intensities aligned in a raster-scan order, and the FFT power spectrum. The size of the SOM competition layer was set to 6 × 6 units by trial and error and 323000 iteration were performed. The neighborhood distance at iteration t is given by $2(1 - t/323000)$.

Figure 5 shows an example feature map obtained after SOM learning. Cells enclosed by thick frames contain more than 50% true solutions and hence were labeled true candidate cells. The remainder are labeled false cells. Figure 5 shows a cluster of true candidate cells in the upper right corner. To examine SOM classification ability, the true positive rate, false positive rate, and averaged positive rate from 5 trials were calculated for each case.

Table 2 summarizes the results of both learning from the study data and clustering for the test data using the map with the best true positive rate. The result shows that learning using image vectors was more accurate than that using the FFT power spectrum and classified candidates with an average positive rate of 89.7%, which is much higher compared with the value of 78.4% for FFT power spectrum.

The most accurate map classified the unknown data with an average positive rate of 86%. This indicates that the utilization of SOM feature map learned from image vectors is an effective technique for the classification of solution candidates. It should also be noted that selecting the most suitable map from the trials is important to improve classification accuracy because performance varied significantly according to the initial conditions.

5. APPLICATION TO OTHER IMAGERY

The effectiveness of the proposed technique for crater detection was examined using imagery that had not been used in the optimization process. In addition, multiresolution images were used to handle craters with a wide range of sizes. Since the radius of target crater ranged from 9 to 18 pixels, it was possible to detect craters with a radius up to 72 pixels using the multiresolution images of three levels.

Figure 6 shows examples of detection and screening results for four images. It can be seen that detection ability is improved significantly even without manual operations. Unfortunately, the achieved detection rate is not sufficient for scientific analysis, thus other detection methods should be considered for some groups and the selection of circle detection modules should also be included in future work. However, it is suggested that the framework presented in this study itself is suitable for applications in which specific features are extracted from a large set of imagery of varying quality.

6. CONCLUSIONS

A technique for mining features from sets of large scale of optical imagery of varying quality has been proposed. The original images were grouped according to spatial frequency patterns, and optimized parameter sets and noise reduction methods were used in the detection process. Furthermore, to improve solution accuracy, false solutions were excluded using SOM feature map that learned true and false solution patterns from a large number of crater candidates. Application of the extracted information to new imagery verified effectiveness of this approach.

The accuracy of detection achieved in this study, however, is not sufficient in comparison with the requirements for scientific analysis and it is necessary to include other detection methods in the future work. However, we believe combining automated abstraction and summarization processes with the accurate manual techniques is crucial for the development of an accurate scientific data mining system. The proposed technique is applicable to various applications in which specific features need to be extracted from large-scale of imagery databases.

7. ACKNOWLEDGEMENTS.

The authors are grateful to Syuta Yamanaka and Hisashi Yokogawa for their past contribution. This research is supported by a grant-in-aid for intensive research (A) (1) (Project 1130215) from the Ministry of Education, Science, and Culture of Japan.

8. REFERENCES

[1] M. C. Burl, L. Asker, P. Smyth, U. Fayyad, P. Perona, L. Crumpler, and J. Aubele. Learning to recognize

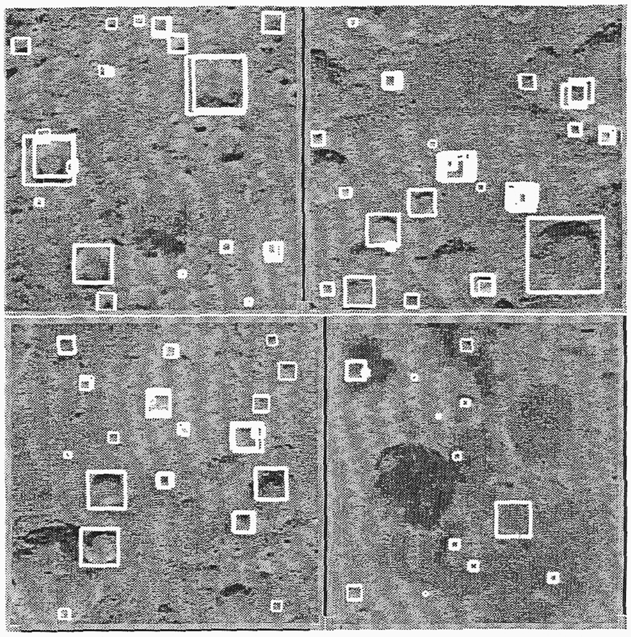


Figure 6: Example of crater detection on new imagery.

volcanos on venus. *Machine Learning*, 30(2/3):165–195, April 1998.

[2] M. C. Burl, C. Fowlkes, and J. Roden. Mining for image content. In *Systems, Cybernetics, and Informatics / Information Systems: Analysis and Synthesis, (Orlando, FL)*, July 1999.

[3] R. O. Duda and P. E. Hart. *Pattern Classification and Scene Analysis.* John Willey and Sons, New York, 1973.

[4] U. M. Fayyad, G. S. Djorgovski, and N. Weir. Automatic the analysis and cataloging of sky surveys. In *Advances in Knowledge Discovery and Data Mining, AAAI Press/ The MIT Press, Menlo Park*, pages 471–493, 1996.

[5] D. E. Goldberg. *Genetic Algorithm in search optimization and machine learning.* Addison Wesley, Reading, 1989.

[6] T. Hamada, A. Shimizu, J. Hasegawa, and J. Toriwaki. A method for automated construction of image processing procedure based on misclassification rate condition and vision expert system impress-pro (in japanese). *Transaction of Information Processing Society of Japan*, 41(7):1937–1947, July 2000.

[7] C. J. Hilditch. Linear skeletons from square cupboards. In *Machine Intelligence 4, Edinburgh Univ. Press, Edinburgh*, pages 403–420, 1969.

[8] R. Honda, O. Konishi, R. Azuma, H. Yokogawa, S. Yamanaka, and Y. Iijima. Data mining system for planetary images - crater detection and categorization -. In *Proceedings of the International Workshop on Machine Learning of Spatial Knowledge in conjunction with ICML, Stanford, CA*, pages 103–108, July 2000.

[9] R. Honda, H. Takimoto, and O. Konishi. Semantic indexing and temporal rule discovery for time-series satellite images. In *Proceedings of the International Workshop on Multimedia Data Mining in conjunction with ACM-SIGKDD Conference, Boston, MA*, pages 82–90, August 2000.

[10] P. V. C. Hough. Method and means for recognizing complex patterns. *U. S. Patent*, 3069654, 1962.

[11] T. Kohonen. *Self-Organizing Maps 2nd ed.* Springer-Verlag, Berlin Heidelberg, 1997.

[12] T. Nagase, T. Agui, and H. Nagahashi. Pattern matching of binary shapes using a genetic algorithm (in japanese). *Transaction of IEICE*, 76-D-II(3):557–565, March 1993.

[13] P. Smyth, M. C. Burl, and U. Fayyad. Modeling subjective uncertainty in image annotation. In *Advances in Knowledge Discovery and Data Mining, AAAI Press/ The MIT Press, Menlo Park*, pages 517–539, 1996.

[14] A. S. Szalay, P. Z. Kunszt, A. Thakar, J. Gray, D. Slutz, and R. J. Brunner. Designing and mining multi-terabyte astronomy archives: The sloan digital sky survey. In *Proceeding ACM-SIGMOD International Conference on Management of Data, Dallas TX*, pages 451–462, May 2000.

[15] T. Watanabe and T. Shibata. Detection of broken ellipse by the hough transforms and multiresolutional images (in japanese). *Transaction of IEICE*, 73-D-II(2):158–166, February 1990.

Support Vector Machines and Kernel Fisher Discriminants: A Case Study using Electronic Nose Data

Dennis DeCoste and Michael C. Burl
Machine Learning Systems Group
Jet Propulsion Laboratory
California Institute of Technology
4800 Oak Grove Drive; Pasadena, CA 91109
{decoste,burl}@aig.jpl.nasa.gov

Alan Hopkins and Nathan S. Lewis
Department of Chemistry
California Institute of Technology
M/S 127-72
Pasadena, CA 91125
nslewis@caltech.edu

ABSTRACT

Kernel methods provide a promising new family of algorithms for machine learning and data mining applications. In particular, kernel-based nonlinear classifiers such as support vector machines (SVMs) and kernel fisher discriminants (KFDs) have been found to work well in practical problems. In addition, there are methods for training these algorithms on large-scale data sets making them very suitable for use in data mining. In this paper, we evaluate the performance of SVMs and KFDs on a dataset generated with a conducting polymer composite-based electronic nose. The ability of SVM and KFD classifiers to correctly identify the functional class (category) of a chemical based on its electronic nose signature is evaluated and compared against other more traditional methods, including nearest neighbors and linear Fisher discriminants. Tradeoffs between the different kernel methods and performance relative to more traditional methods are discussed.

Keywords

support vector machines, kernel Fisher discriminant, classification, electronic nose

1. INTRODUCTION

Arrays of polymer films embedded with conductive or resistive material have attracted significant attention as "electronic noses." Unlike traditional "lock-and-key" approaches to vapor sensing, in which a detector is very specific to a particular analyte, the polymer-based detectors used here are broadly-tuned so that a given detector responds to many vapors and a single vapor causes a response in many detectors. Only by analyzing the pattern of responses across the array of detectors can specific analytes be identified or discriminated from chemically similar compounds. As part of an ongoing scientific research project between JPL and Caltech [2], we have been studying the suitability of kernel-based methods for various classification tasks involving data from the electronic nose. In this paper, new results using support vector machines (SVMs) and kernel Fisher discriminants (KFD) to learn to predict the category (e.g., alcohol or hydrocarbon) of a previously unseen (unsniffed?) chemical are presented. We will discuss how these results illustrate some of the practical decisions and tradeoffs required to apply SVM and KFD methods and contrast their performance against other traditional methods (i.e. nearest-neighbors and linear Fisher discriminants).

2. ELECTRONIC NOSE

The Caltech electronic nose consists of an array of polymer films embedded with conductive or resistive material. Sorption of a vapor into the polymer films causes physical swelling, which leads to a change in the DC electrical resistance of the film. The DC resistance across each of the films in the array is sampled at approximately uniformly-spaced sample times. The resistance values are digitized with an A-to-D converter. For the experiments reported here, the raw time-series data were converted to vector form by computing the relative change in resistance in each channel compared to the pre-exposure baseline. The raw time-series response of the electronic nose to a given analyte thus becomes a d-dimensional vector where d is the number of channels (polymer films).

All analyte exposures were performed using a computer-controlled vapor generation and control system that regulates the identity, concentration, exposure time, and flow rate of the analyte above the detectors [20]. Between exposures, clean air is passed through the system to remove any residue from the previous exposure. Analytes are presented to the system in a randomized order to prevent biases in the results. For a broad range of concentrations and analytes, the electronic nose arrays behave like a linear system. Increasing or decreasing the concentration of an analyte produces a proportional increase or decrease in the signature, and the response to mixtures of analytes is approximately the weighted average of the response to the individual analytes [20].

An interesting study, which we have recently undertaken (preliminary results using nearest neighbor classification appeared in [2]), involves learning to classify analytes into the appropriate functional group based on their electronic nose signature. Five functional groups or chemical families (alcohols, alkyl halides, aromatics, hydrocarbons, and esters) were used for our experiments. Within each class, 15 members were chosen for a total of 75 different chemicals. These were presented to an electronic nose containing 40 polymer-based sensors (two copies of 20 different polymers). The analytes were presented to the nose in groups of eight because the physical setup of the gas dispensation system has eight bubblers. A total of 80 sniffs (ten of each analyte) in randomized order was obtained from each set of eight and then a new set of eight analytes was swapped in. Each group of eight analytes contained two members of four classes so that temporal effects would not bias the results (e.g., if all alcohols were sniffed in the morning and the temperature was cooler then, it might introduce an artificial bias into the separability of alcohols from the other classes). Each sniff produced a multivariate time series which was converted to vector form. Responses of polymer "twins" (duplicates of the same polymer type) were averaged to produce 20-dimensional vectors for each sniff. This data was then used by various kernel-based and traditional classification algorithms in a leave-one-chemical-out (LCO) cross-valdiation. Note that for a given test example, all other sniffs of the same compound were sequestered from the training set. In other words, we wanted to determine if a sniff of methanol could be used to classify it as an alcohol *without* having previously smelled methanol, but perhaps having smelled ethanol, butanol, cyclopentanol, etc.

3. KERNEL METHODS

Recently, many traditional linear methods have been generalized to corresponding nonlinear forms using *Mercer kernels*. Examples include Principal Component Analysis [19], k-means clustering [18] nearest-neighbors [18], and Fisher discriminants [14]. Further, completely new kernel-based methods such as SVM classification [1] and SVM regression [21] have been introduced.

Consider an ℓ-by-D data matrix (X) of examples. A (Mercer) kernel $K(x_i, x_j)$ implicitly projects the two given examples from D-dimensional input space into some (possibly infinite-dimensional) feature space and returns their dot product in that feature space. That is, it computes

$$K(x_i, x_j) \equiv \phi(x_i) \cdot \phi(x_j) \equiv \phi(x_i)' \phi(x_j), \qquad (1)$$

for some mapping function ϕ, but without explicitly computing the coordinates of the projected vectors. In this way, kernels allow large non-linear feature spaces to be explored while avoiding the curse of dimensionality.

The simplest kernel is the *linear* kernel, implemented as a simple dot product:

$$K(u, v) = u \cdot v \equiv \sum_{i=1}^{d} u_i \cdot v_i. \qquad (2)$$

As explained later, kernel methods give the same results as their traditional linear equivalents when linear kernels are used, but will typically be much slower. This cost arises from operating on some matrices of size ℓ-by-ℓ that are only of size D-by-D in traditional linear methods.

The *polynomial* kernel is defined by a non-linearly squashed dot product of the following form:

$$K(u, v) = (u \cdot v + r)^d, \qquad (3)$$

with polynomial degree parameter d. Varying the continuous offset parameter r changes the relative weighting of the (implicit) terms in the non-linear polynomial feature space. We will refer to instances of this kernel as "POLY d r".

One of the most popular kernels is the *radial basis function* (RBF) kernel:

$$K(u, v) = e^{\frac{-||u-v||^2}{2\sigma^2}}, \qquad (4)$$

with variance parameter σ, giving another non-linear squash of the dot product of the two examples. [1] We will refer to instances of this kernel as "RBF g", where $g = \frac{1}{2\sigma^2}$.

In this paper we will focus on two specific kernel methods, SVMs and KFDs, as described below.

3.1 Support Vector Machine (SVM)

Given an ℓ-by-ℓ kernel matrix K (computed from the ℓ-by-D data matrix (X)), an ℓ-by-1 labels vector (y), and a "soft margin" regularization parameter ($C > 0$), training a binary SVM classifier traditionally consists of the following Quadratic Programming (QP) dual formulation:

minimize:

$$\tfrac{1}{2} \textstyle\sum_{i=1}^{\ell} \sum_{j=1}^{\ell} \alpha_i \alpha_j y_i y_j K(x_i, x_j) - \sum_{i=1}^{\ell} \alpha_i$$

subject to:

$$0 \leq \alpha_i \leq C, \ \textstyle\sum_{i=1}^{\ell} \alpha_i y_i = 0,$$

where ℓ is the number of training examples, y_i is the label (+1 for positive example, -1 for negative) for the i-th training example (x_i), and $K(x_i, x_j)$ denotes the value of the kernel function for i-th and j-th examples of X.

Note that once the kernel matrix is computed, the SVM itself is independent of both the input dimensionality (D) and the (implicit) feature dimensionality (in kernel space). In this way, SVMs are said to overcome the curse of dimensionality.

The vector of alphas α (of length ℓ) is the solution to the above QP problem. Of significant practical benefit is that there are no local optima for this QP (unlike, say, neural networks).

Widely-used decomposition methods, such as SMO [16] and SVM^{light} [9], typically train SVMs (i.e. solve the above QP) in roughly O(ℓ^2) time and sub-quadratic space (by computing kernel elements only as required). In our experiments, we use our own implementation [3] of SMO, based on [10].

The SVM output classification $F(x)$, for any new example x, can be computed as:

$$G(x) = \sum_{i=1}^{\ell} \alpha_i y_i K(x, x_i), \qquad (5)$$

[1] Where 2-norm defined as $||u - v||^2 \equiv (u \cdot u - 2u \cdot v + v \cdot v)$.

$$F(x) = sign(G(x) - b), \tag{6}$$

The SVM weights (α) over the examples are often rather sparse (typically roughly 5% — 20% are non-zero), making the above output summations somewhat faster in practice than shown above. The special examples x_i for which $0 < \alpha_i \leq C$ are called the *support vectors* (SVs). Retraining the SVM using only the SVs would result in the same α solution.

Let SV^+ represent the set of positive support vector examples and SV^- represent the set of negative SV examples. Similarly, define their corresponding "in-bounds" subsets IN^+ and IN^-, for which $0 < \alpha_i < C$. As is common practice, we compute the scalar bias (b) as midway between the mean of G over IN^+ and the mean of G over IN^-.

A SVM maximizes the *margin* distance between the nearest positive and negative examples (in kernel feature space), which has been shown to lead to excellent generalization performance in many domains [7], for much the same reasons as the similar success of boosting methods [6].

3.2 Kernel Fisher Discriminant (KFD)

The classic linear Fisher discriminant (LFD) for binary classification [5] finds the projection weights (w) that map the data X onto a line such that along that line within-class variance is minimized while between-class variance is maximized.

3.2.1 LFD

Specifically, LFD maximizes the following score J:

$$maximize: \quad J(w) = \frac{w' \; S_B \; w}{w' \; S_W \; w}. \tag{7}$$

The between (S_B) and and within (S_W) components of J are defined as:

$$S_B = (m^+ - m^-)(m^+ - m^-)', \tag{8}$$

$$S_W = \sum_{x_i \in X-} (x_i - m^-)(x_i - m^-)' + \sum_{x_i \in X+} (x_i - m^+)(x_i - m^+)', \tag{9}$$

where

$$m^- = \frac{1}{\ell^-} \sum_{x_i \in X-} x_i, \quad m^+ = \frac{1}{\ell^+} \sum_{x_i \in X+} x_i, \tag{10}$$

are the D-dimensional mean vectors for the negative (X^-) and positive (X^+) examples, respectively.

The D-dimensional projection weights can be computed in closed-form using:

$$w = S_W^{-1}(m^- - m^+). \tag{11}$$

The LFD classification $f(x)$ for example x is given simply by:

$$g(x) = x'w, \qquad f(x) = sign(g(x) - b), \tag{12}$$

where b is a threshold (typically determined on the assumption that the class-conditional denisities of the projected data are Gaussian).

3.2.2 KFD

By substituting $\phi(x_i)$ for each x_i in LFD,, denoting each resulting $\phi(x_i) \cdot \phi(x_j)$ term as kernel element $K(x_i, x_j)$, and using some algebraic simplifications, we get KFD (e.g. [13]):

$$maximize: \quad J(\alpha) = \frac{\alpha' \; Z_B \; \alpha}{\alpha' \; Z_W \; \alpha}. \tag{13}$$

$$Z_B = (\mu^+ - \mu^-)(\mu^+ - \mu^-)', \tag{14}$$

$$Z_W = KK' \tag{15}$$

where

$$\mu^- = \frac{1}{\ell^-} K \; 1^-, \quad \mu^+ = \frac{1}{\ell^+} K \; 1^-, \tag{16}$$

act like (ℓ-dimensional) "mean" vectors [2] and K is the ℓ-by-ℓ kernel matrix with elements:

$$K_{ij} = k(x_i, x_j). \tag{17}$$

The α are computed in closed-form (analogous to w in LFD):

$$\alpha = Z_W^{-1}(\mu^- - \mu^+). \tag{18}$$

The projection weights in feature space themselves are then given by:

$$W = \sum_{i=1}^{\ell} \alpha_i \phi(x_i) \tag{19}$$

The KFD classification $F(x)$ for example x follows the same form as for SVMs, except that α_i is no longer restricted to be non-negative and labels y_i no longer appear:

$$G(x) = \phi(x)'W = \sum_{i=1}^{\ell} \alpha_i K(x, x_i), \tag{20}$$

where W is the (implicit) weight vector in kernel feature space, and

$$F(x) = sign(G(x) - b). \tag{21}$$

For a linear kernel, the D-dimensional weights (w) of LFD can be recovered, by "weight folding" KFD's ℓ-dimensional (α):

$$w = \quad W = \sum_{i=1}^{\ell} \alpha_i \phi(x_i) \quad = \sum_{i=1}^{\ell} \alpha_i x_i. \tag{22}$$

This shows that KFD with the linear kernel gives the identical weights as LFD (but is much slower to train).

As in SVM's, some form of regularization for KFD is required in practice. One common approach, which we employ in our experiments here, is to add some regularization scalar to the diagonal of the Z_W. This also prevents inversion problems when Z_W is nearly singular.

[2] 1^+ is the ℓ-by-1 vector which contains ones where the labels vector y has 1's and contains zeros elsewhere. 1^- is similar, but contains ones where target y has -1's.

One remaining issue for KFD is how to compute the threshold bias (b). One approach ([15]), which we employ here is to train a linear SVM, using the projected KFD outputs as (1-dimensional) training data, and use the bias computed by the SVM. Other approaches are described in [13].

In contrast to SVMs, KFDs have recently been shown [13] to roughly maximize the *average margin*, i.e. the distance between the centers of the positive and negative data once they are projected on the Fisher line.

4. CLASSIFICATION EXPERIMENTS

4.1 Nearest Neighbors

For baseline comparisons, we repeat here earlier results [2], using 1-nearest-neighbors. The Euclidean distances of the test example from all members of the reference library were computed. The functional class label of the closest member of the reference library was taken to be the class label of the test example. *For a given test example, all other sniffs of the same compound were sequestered from the reference library.* In other words, we wanted to determine if a sniff of methanol could be used to classify it as an alcohol *without* having previously smelled methanol, but perhaps having smelled ethanol, butanol, cyclopentanol, etc. A confusion matrix showing the results of this experiment is given in Table 1. The first row shows that all members of the alcohol family were correctly classified as alcohols. The second row shows that 83% of the members of the alkyl halide family were correctly classified, with 6.9% of the members confused as aromatics, 0.6% confused as hydrocarbons, and 9.4% confused as esters. Overall, the average correct classification percentage is 77%.

4.2 Handling Multiple Classes

There are several ways to handle multiple (k) classes (in our case, k=five) using binary classifiers. The two that we have explored can be described as "one-vs-rest" and "pair-wise voting".

In one-vs-rest, one learns k classifiers, each deciding if an example is of that class or not. One decides which of the k classes it is by finding which of the k classifiers has the strongest positive output.

In pair-wise voting, one learns $k(k-1)/2$ classifiers, for each pair-wise contest. If a single class unanimously wins all pair-wise contests for an example versus each of the other $k-1$ classes, then its label is assigned to the example. If a unanimous decision cannot be reached, it is treated as a "punt" (i.e. no classification is made).

Due to computational expense, to date we have only tried one-vs-rest for our kernel methods. We have tried both ways for LFD. The results are presented below.

4.3 Kernel Methods

Kernel methods require model selection to select appropriate kernels – both type (e.g. polynomial vs RBF) and parameters (i.e. poly degree or RBF variance level). Ideally, one would do model selection search (e.g. via cross-validation) for each leave-one-chemical-out in our experiment. However, current techniques make that too costly, especially for KDA since no efficient model selection methods have yet been formulated (see [4] and [11] for some recent methods for more efficient SVM model selection).

Thus, we simply selected kernels for SVM and KFD based on which worked well when randomly partitioning the data set into training and validation sets. This is likely to be suboptimal, since we thus selected one kernel to use regardless of which chemical class is being left out in turn in the final test experiment.

It is also possible that this approach is slightly contaminated, since some final test chemicals occur in training sets during this process. However, we only did this model selection search to determine "reasonable" kernels to use. The actual model weights in that feature space (e.g. the SVM's or KFD's α) are trained in the final test experiments with no knowledge of the hold-out chemicals.

Tables 2 and 3 show the results for the best kernel selected for SVMs and KFDs, respectively.

Note that we found KFD worked best with the unnormalized polynomial kernel ($K(u,v) = (u \cdot v + 10)^3$) whereas SVM worked best with the normalized version: $K(u,v) = \frac{1}{2^3}(u \cdot v + 1)^3$, where all u and v are 2-normed (unit length) versions of the original data. This result is consistent with observations made elsewhere that SVMs seem to work best when feature vectors are normalized [8]. KFD apparently does not benefit from such normalization, due to the way Fisher discriminants use the covariance matrix explicitly.

The SVM appears to work significantly better. We believe part of the reason may be because one-versus-rest approaches to multi-class problems are not particularly suitable for Fisher discriminants. The next section shows that indeed, at least for LFD, pair-wise classifiers seems to be better than one-versus-rest. In fact, the one-vs-rest LFD result shows that our KFD result is no better than that.

4.4 Linear Fisher Discriminants

Tables 4 and 5 show the results for the linear Fisher discrimnation, using one-vs-rest and pairwise approaches, respectively, on our multi-class classification problem.

5. CONCLUSIONS

Our results show that kernel methods offer some promise for challenging real-world tasks such as our chemical functional class problem. However, we are still working on several important issues. One is conducting more comprehensive model selection to more accurately (and fairly) determine the best kernels to use for each chemical hold-out experiment. We are also studying the tradeoffs of one-vs-rest and pair-wise approaches to multi-class problems such as ours. And we are looking into the best way to use soft (probabilistic) target labels. For example, scientists recently provided us with fractional assignments of the chemicals to the five groups. We suspect that the current ceiling of test performance (near 85%) may be exceeded once we more fairly account for the fast that many of these chemicals do not fall exclusively in only one of the five groups.

6. REFERENCES

[1] C. Burges. A tutorial on support vector machines for pattern recognition. *Data Mining and Knowledge Discovery*, 2(2), 1998.

[2] M. C. Burl, S. Briglin, B. Doleman, A. Hopkins, A. Matzger, D. N. Ortiz, A. Schaffer, S. Upchurch, T. Vaid, and N. S. Lewis. Mining the detector responses of a conducting polymer composite-based electronic nose. In *First SIAM Int. Conf. on Data Mining*, April 2000.

[3] D. DeCoste and B. Schölkopf. Training invariance support vector machines. *Machine Learning*, 2001. In press.

[4] D. DeCoste and K. Wagstaff. Alpha seeding for support vector machines. In *International Conference on Knowledge Discovery and Data Mining (KDD-2000)*, August 2000.

[5] R. O. Duda and P. E. Hart. *Pattern Classification and Scene Analysis*. Wiley, New York, 1973.

[6] A. Grove and D. Schuurmans. Boosting in the limit: Maximizing the margin of learned ensembles. In *Proceedings of the Fifteenth National Conference on Artificial Intelligence*, pages 692–699, 1998.

[7] I. Guyon. Online SVM application list, 2000. (See http://www.clopinet.com/isabelle/Projects/SVM/applist.html.).

[8] R. Herbrich and T. Graepel. A PAC-bayesian margin bound for linear classifiers: Why SVMs work. *Advances in Neural Information System Processing (NIPS) 13*, 2001.

[9] T. Joachims. Making large-scale support vector machine learning practical, 1999. In *Advances in Kernel Methods: Support Vector Machines* [17].

[10] S. Keerthi, S. Shevade, C. Bhattacharyya, and K. Murthy. Improvements to Platt's SMO algorithm for SVM classifier design. Technical Report CD-99-14, Dept. of Mechanical and Production Engineering, National University of Singapore, 1999.

[11] J.-H. Lee and C.-J. Lin. Automatic model selection for support vector machines. Technical report, Dept. of Computer Science and Information Engineering, National Taiwan University, Taipei, Taiwan, November 2000. Implementation at [12]. Online at http://www.csie.ntu.edu.tw/ cjlin/papers/modelselect.ps.gz.

[12] J.-H. Lee and C.-J. Lin. LOOMS: Leave-one-out model selection for support vector machines, 2000. Software at http://www.csie.ntu.edu.tw/ cjlin/looms/.

[13] S. Mika, G. Rätsch, and K.-R. Müller. A mathematical programming approach to the kernel fisher algorithm. In *Advances in Neural Information Processing Systems (NIPS) 13*, 2001.

[14] S. Mika, G. Rätsch, J. Weston, B. Schölkopf, and K.-R. Müller. Fisher discriminant analysis with kernels. In Y.-H. Hu, J. Larsen, E. Wilson, and S. Douglas, editors, *Neural Networks for Signal Processing IX*, pages 41–48. IEEE, 1999.

[15] S. Mika, A. Smola, and B. Schölkopf. An improved training algorithm for kernel Fisher discriminants. In T. Jaakkola and T. Richardson, editors, *Artificial Intelligence and Statistics*, pages 98 – 104, San Francisco, CA, 2001. Morgan Kaufmann. Also: Microsoft Research TR-2000-77.

[16] J. Platt. Fast training of support vector machines using sequential minimal optimization, 1999. In *Advances in Kernel Methods: Support Vector Machines* [17].

[17] B. Schölkopf, C. Burges, and A. Smola. *Advances in Kernel Methods: Support Vector Machines.* MIT Press, Cambridge, MA, 1999.

[18] B. Schölkopf, A. Smola, and K. Müller. Nonlinear component analysis as a kernel eigenvalue problem. Technical report no. 44, Max-Planck-Institut for Biologische Kybernetik, Tübingen, Dec 1996.

[19] B. Schölkopf, A. Smola, and K. Müller. Nonlinear component analysis as a kernel eigenvalue problem. *Neural Computation*, 10(5):1299–1319, 1998.

[20] E. Severin, B. Doleman, and N. Lewis. An investigation of the concentration dependence and response to analyte mixtures of carbon black-insulating organic polymer composite vapor detectors. *Anal. Chem.*, 72:658–668, 2000.

[21] A. Smola and B. Schölkopf. A tutorial on support vector regression. *Statistics and Computing*, 2000.

1NN: 0.77	alcohol	alkyl halide	aromatic	hydrocarbon	ester
alcohol	1.000	0.000	0.000	0.000	0.000
alkyl halide	0.000	0.831	0.069	0.006	0.094
aromatic	0.000	0.267	0.527	0.200	0.007
hydrocarbon	0.037	0.019	0.213	0.688	0.044
ester	0.047	0.147	0.000	0.018	0.788

Table 1: LCO confusion matrix for 1NN classification of compounds into functional classes. Each functional class contained fifteen compounds with 10 or in a few cases 20 sniffs each. For a given test example, all other examples of the same compound were withheld from the reference library (i.e. LCO = Leave-Chemical-Out). Average correct LCO classification rate = 0.77.

SVM: 0.82	alcohol	alkyl halide	aromatic	hydrocarbon	ester
alcohol	1.000	0.000	0.000	0.000	0.000
alkyl halide	0.013	0.688	0.062	0.000	0.237
aromatic	0.000	0.113	0.713	0.120	0.053
hydrocarbon	0.031	0.062	0.050	0.856	0.000
ester	0.059	0.100	0.000	0.000	0.841

Table 2: LCO confusion matrix for SVM trained one-vs-rest (kernel='poly 3 .1' C=100). Average correct rate = 0.821, total runtime = 868.9 secs. (No-holdout training rate = 0.921.)

KFD: 0.79	alcohol	alkyl halide	aromatic	hydrocarbon	ester
alcohol	1.000	0.000	0.000	0.000	0.000
alkyl halide	0.000	0.606	0.062	0.131	0.200
aromatic	0.067	0.047	0.687	0.133	0.067
hydrocarbon	0.025	0.050	0.075	0.850	0.000
ester	0.059	0.124	0.006	0.000	0.812

Table 3: LCO confusion matrix for KFD trained one-vs-rest (kernel='POLY 3 10'). Average correct rate = 0.792, total runtime = 1919.8 secs. (No-holdout training rate = 0.935.)

LFD: 0.80	alcohol	alkyl halide	aromatic	hydrocarbon	ester
alcohol	0.931	0.056	0.000	0.000	0.013
alkyl halide	0.000	0.719	0.081	0.075	0.125
aromatic	0.007	0.027	0.713	0.133	0.120
hydrocarbon	0.031	0.031	0.062	0.875	0.000
ester	0.076	0.165	0.006	0.000	0.753

Table 4: LCO confusion matrix for LFD trained one-vs-rest (i.e. KFD with kernel='linear'). Average correct rate = 0.799, total runtime = 17.4 secs. (No-holdout training rate = 0.927.)

LFDp: 0.84	alcohol	alkyl halide	aromatic	hydrocarbon	ester
alcohol	0.988	0.000	0.000	0.000	0.000
alkyl halide	0.000	0.831	0.069	0.006	0.000
aromatic	0.000	0.067	0.660	0.133	0.013
hydrocarbon	0.000	0.006	0.138	0.831	0.019
ester	0.053	0.006	0.000	0.006	0.871

Table 5: LCO confusion matrix for LFD with pairwise voting. Average correct rate = 0.836. Note: rows do not sum to 1 because deadlocks between the different pairwise classifiers are treated as "punts".

Time-Invariant Sequential Association Rules: Discovering Interesting Rules in Critical Care Databases

Jafar Adibi, Wei-Min Shen
Information Sciences Institute, University of Southern California
4676 Admiralty Way, Marina del Ray, CA 90292
{adibi,shen}@isi.edu

ABSTRACT

Discovering patterns in sequences of events has been an area of active research in Artificial Intelligence and Data Mining. Many existing techniques which generate sequential association rules have two major problems: they either produce too many rules or they cannot discover rules that have high *confidence*, but weak *support*. Both cases make manual inspection and analysis very difficult. The focus in this body of work is on discovering such rules in a recognized group of special databases, in which data are not uniformly distributed and exhibit self-similarity and fractal dimensionalities. We introduce, study and analyze a group of sequential association rules as time-invariant and self-similar association rules. We provide a formalism to discover such rules through the discovery of association rules with the high degree of *confidence* and *support*. Time-invariant and self-similar association rules has been investigated in the context of Critical Care database which has been collected during past 15 years at the King Drew Medical Center and Harbor UCLA Hospital. Even thought the obtained result is in early stage but they are encouraging and we would like to apply this technique to other synthetic and real databases in the future.

Keywords

Association Rules, Self-Similarity, Time Invariance

1. INTRODUCTION

Much of the existing data mining techniques have been focused on designing efficient methods to mine knowledge and patterns from databases. Sequential association rule is one the most well known form of extracted knowledge. Instead of statistical methods which are looking for a global model for data, association rules mainly find local patterns. An association rule is in the form ($P \Rightarrow Q$), where P and Q are sets of attributes, meaning that in the rows of the database where the attributes in P have true value, also the attributes in Q tend to have true value. Association rules define with two major parameters: *support* and *confidence*. The *support* of a given rule is the ratio of the records having true values for the attributes of ($P \cup Q$) to number of all records, whereas the *confidence* of that rule is the ratio of the number of records having true values for attributes of ($P \cap Q$) to the number of records having true values for attributes of P.

The main approach for mining association rule in general derives by Agrawal et al [4] call a-priori, which exploits the support requirements for association rules. The key observation is that if a set of attributes appears in a fraction of the tuples, then any subset of such set also appears in a fraction of the tuples. Variants and enhancements of this approach underlie essentially all known efficient algorithms for computing association rules or their variants. The general algorithm mainly works with the *support* level and *confidence* requirement plays no role in the algorithm, and is completely ignored until the end of the discovery loop when high-supported sets are screened for high *confidence*.

Many existing techniques which generate association rules are facing two major problems: they often either produce too many rules and/or they cannot find *interesting rules*. By *interesting rules* we refer to those rules which have extremely high confidence, but for which there is weak support. Both cases make manual inspection and analysis very difficult.

This work is motivated by the open question of discovering *interesting rules*. For example, in medical treatment domain, the standard association rule algorithms may be useful for extracting patterns with high *support* such as

> *"if treatment A applies to patient in first hour of admission, she recovers in less than 8 hours"*,

but are essentially useless for discovering rules such as

> *"if treatment A applies to patient after 96 hours, she may recovers after4 days"*,

because there are only a few patients who received the treatment after the 96 hours following the admission procedure.

There are two possible objections to removing the support requirement form the discovery process[6]. First, this may cause an explosion in the number of rules that are produced and make it difficult for a user to distinguish the rules of interest or take huge amount of time to discover such rules. Second, it may be argued that rules of low support are uninteresting. While this might be true in the classical market-basket applications, there are many applications where it is essential to discover such rules of extremely high *confidence* without enough *support*. For many

Conference 'KDD-01, August, 2001, San Francisco, CA.
Copyright 2001 ACM 1-58113-000-0/00/0000...$5.00.

scientific databases *interesting rules* are indeed very crucial and does not happen so often. To name a few: medical database, financial datasets including transaction databases, identifying identical or similar documents or web pages, identifying similar vectors in high-dimensional spaces and collaborative filtering [6] are examples in which a rule with low-support and high confidence is very crucial . Some of these applications consists of a sparse table and the goal is to identify column pairs that appear to be similar, without any support requirement. In addition, detecting causality is another important form in data mining, where it is important to discover associated fields, but there is no notion of support [17].

The implicit assumption in some of the studies on sequential association rules is that the data is uniformly distributed and attributes are independent from each other. This assumption basically implies that a-priori like algorithms do not use of the data structure, shape and characteristics. However, real data sets disobey these assumptions. A recognized groups of data typically are skewed and exhibit fractal dimensionalities. For instance, most of the biological systems contain self-similar structures that are made through recurrent processes. In these databases, the information and embedded complexity are hierarchical. In addition, they are self-similar and/or contain self-similar structures and/or have been generated through recurrent processes at least up to a certain level. Many physical systems contain a form of functional self-similarity that owes its richness to recursion. Human brains, economic markets, musical notes, network data also create enormously complex behavior that is much richer than the behavior of the individual component units. New findings in different branches of science and technology also show the presence of self-similarity in different domains. To name a few: medical diagnosis (physician treatment and patient response), Robot navigation (robot move and environment response to robot sensors) and Network behavior monitoring (packet transmission, switch behavior and network response) are examples of such environments.

The main goal of this paper is to provide a novel technique to address above-mentioned problems for a special form of temporal databases which shows self-similarity up to a certain degree. In general, self-similarity or long range dependence refers to observation of similar patterns when a discrete or continuous time process is scaled in time. The process in larger scale is a copy of itself in smaller time scales. We employ such idea for self-similar databases. The presence of a rule in small scale is a copy of itself in larger scale and vise versa. As regular a-priori algorithms discover rules in smaller scale with enough *user define support* we can discover rules in larger scale through the discovered rules in smaller scale even if they do not have enough support.

While there have been much effort on observing self-similar structures in scientific databases and natural structures, there is few work on using self-similar structure and fractal dimension for data mining, predictive modeling and forecasting. Among these works, using fractal dimension and self-similarity for managing the dimensionally curse [19], learning association rules [5], application in spatial joint selectivity in databases [9] and Self-Similar Layered HMM for self-similar structures [2] are considerable.

The rest of this paper is organized as follows. In section 2 we explain the problem statement along with notation and definitions. In section 3 we outline related work to this paper. In section 4, we introduce the time-invariant association rules for sequential databases, its definition and properties. In section 5 we discuss or method on discovering time-invariant sequential association rules. Section 6 shows the current result with an experimental finding in Critical Care patient database followed by the future work and conclusions in section 7.

Table 1: Example of a Critical Care database

Patient ID = 1

	Type	Time	Perception/Action		
1	P	8	O_1=101	O_2=1.5	O_3=0
2	P	12	O_1=120	O_2=1.2	O_3=1
3	A	13	X		
4	P	24	O_1=144	O_2=1.5	O_3=1
5	A	26	X		
6	P	28	O_1=106	O_2=1.3	O_3=-1
7	A	32	Y		
8	P	38	O_1=132	O_2=1.2	O_3=1
9	A	44	X		
10	P	55	O_1=101	O_2=1.5	O_3=-1
11	P	67	O_1=108	O_2=1.6	O_3=1
12	A	83	X		
13	P	90	O_1=144	O_2=1.7	O_3=1
14	A	110	Y		
15	P	121	O_1=111	O_2=1.8	O_3=-1
16	A	134	X		
17	P	165	O_1=123	O_2=0.5	O_3=-1
18	A	178	X		
19	A	181	X		
20	P	182	O_1=109	O_2=0.6	O_3=-1
21	P	200	O_1=115	O_2=0.8	O_3=1
22	A	202	X		

Patient ID = 2

	Type	Time	Perception/Action		
1	P	9	O_1=100	O_2=1.0	O_3=0
2	P	13	O_1=121	O_2=1.1	O_3=1
3	A	15	X		
4	P	27	O_1=143	O_2=1.3	O_3=1
5	A	29	X		
6	P	31	O_1=116	O_2=1.7	O_3=-1
7	A	36	Y		
8	P	38	O_1=112	O_2=1.6	O_3=-1
9	A	46	X		
10	P	54	O_1=100	O_2=1.0	O_3=-1
11	P	60	O_1=99	O_2=1.5	O_3=-1
12	A	79	Y		
13	P	92	O_1=108	O_2=1.4	O_3=1
14	P	110	O_1=121	O_2=1.2	O_3=1
15	A	134	X		
16	P	145	O_1=140	O_2=0.9	O_3=1
17	A	153	X		
18	P	162	O_1=111	O_2=0.7	O_3=-1
19	A	177	Y		
20	P	190	O_1=113	O_2=0.8	O_3=1
21	P	204	O_1=120	O_2=1.3	O_3=1
22	A	250	X		

2. PROBLEM STATMENT

We are given a database of sequences $D=\{d_1,d_2, ...,d_n\}$. Each sequence d_i belongs to a patient, customer or in general belongs to an entity and consists of a collection of perceptions and actions. Each item in a sequence is either a perception or an action. Each perception consists of: *entity_id ,perception-time and attribute_set* (*attribute_id, attribute_value*). Each action also consists of *entity_id, action_ time, action_id.* Table 1 shows examples of 2 different patients in a given database. Perceptions O_1, O_2 and O_3 has been observed for all patients across the database and treatments *X* and *Y* has been applied. Grayed rows in Table 1. refer to actions.

2.1 Definition

Given a database *D* of *N* data-sequences, an action set of *C* and Perception set *O*, user-specified *min-gap* and *max-gap* time constraints, the problem of mining interesting sequential patterns is to find all sequences whose support is greater than the *user-specified minimum support* or its confidence is greater than the *user-specified minimum confidence*. Each sequence represents a sequential pattern of perception and action, also called an *interesting sequence*.

Note that the notion of *min-gap* and *max-gap* are different with what Agrawal et al introduced in [3]. In their work the *min-gap* and *max-gap* basically is used for removing noise and outliers in time series matching process. However in our point of view *min-gap* and *max-gap* refers to the length of a sequential pattern in general. If we look at an association rules in form ($P \Rightarrow Q$), where *P* and *Q* are sets of attributes, *min-gap* associate with P while *max-gap associate with P and Q*. The definition of *min-gap* and *max-gap* will be explained later in this chapter.

The main idea is to discover *interesting rules* by scaling the general sequential association rules (mainly those which satisfy *minimum-support*). The notion of scaling which associate with *min-gap* and *max-gap* will explain in section 4.

We do consider quantities of perceptions but do not consider the quantities for actions (for example the dosage of drug in our example): each item is a number for actions and a function of quantity for perceptions. Without loosing generality and for the purpose of better understanding we map actions and perceptions to different set. We denote a sequence by $<s_1,s_2,...s_n>$ in which s_i is an action or perception. Each s_i will be a tuple of quantities and the time (value, time) and it refers as *item.* In general we demonstrate each sequence with $S(i)^{perception}_{start,end}$, in which *i* refers to entity *id* , *perception* stands for the index of the observed perception and *start* and *end* refers to the starting point and ending point of a given sequence. For instance, perception O_3 for patient *ID = 1* in Table.1 will be shown as the following: $S(1)^{3}_{1,10}$ =<(0,8),(1,12),(X,13),(1,24),(X,26),(1,28),(Y,32),(1,38),(X,44),(1,50)>. For the short note we only show the value of perception and action : $S(1)^{3}_{1,10}$ =<0,1,X,1,X,1,Y,1,X,1>. For a specific item *j* in sequence $S(i)^{p}_{s,e}$ we use $S(i)^{p}_{j}$. In addition to show the time of a given item in sequence $S(i)^{p}_{j}$ we use of T($S(i)^{p}_{j}$). For instance in previous example $T(S(1)^{3}_{2})$=12.

2.2 Pattern

We define a pattern as a sequence with additional capabilities to normal sequence. First, a pattern can have a general observation symbol standing for all possible observation or action. We show this symbol by *[A]* for actions and *[O]* for observations and [I] for either action or observation. A pattern also can have * in front of each *item* in sequence. A * represents the zero to infinite number of such item. We denote a pattern with P and the $Lp=|P|$ represent the length of pattern. Patterns only represent the values and do not contain the time of sequence. Table 2 shows a full definition of pattern notation. For instance P_1=<[A]*,1,X,1,Y,[I]*,1> and P_2=<1,[I]*,A,[I]*,1] are simple examples of patterns and $S(i)^{p}_{s,e}$ =<1,0,X,Y,1> and $S(i)^{p}_{s,e}$ =<1,1,X,0,Y,-1,1>are example of sequences which satisfy P_2.

We also associate a *max-gap* (G_{max}) and *min-gap* (G_{min}) with a pattern. G_{max} and G_{min} basically control the length of the sequence. If we consider a sequence in the form of $P \Rightarrow Q$, in which P refers to $S(i)^{p}_{1,|L-1|}$ and *Q* refers to $S(i)^{p}_{|L|}$, G_{max} controls the length of the whole sequence and G_{min} controls the length of the *P* part in $P \Rightarrow Q$. Hence, for a given pattern with defined G_{max} and G_{min}

$$T(S(i)^{p}_{|P|}) - T(S(i)^{p}_{1}) < \mathrm{G_{max}}$$

$$T(S(i)^{p}_{|P-1|}) - T(S(i)^{p}_{1}) > \mathrm{G_{min}}.$$

Table 2: Notation and Definitions

Symbol	Description	Example
item	Either Action or Perception	X,1
A	Actions	X
O_i	Perceptions	-1
[A]	Any kind of actions	Y
[O]	Any kind of perceptions	1
[I]	Any item (actions or perceptions)	X,1
Y*	Unlimited number of perception Y	Y,Y,Y
[A]*	Unlimited number of actions	X,Y,X
[O]*	Unlimited number of perceptions	1,1,-1
[I]*	Unlimited number of actions or perceptions	X,1,Y,-1

Rephrasing the problem statement, we are looking for all frequent patterns $<s_1,s_2,...s_n>$ when their confidence is grater than user defined *minimum-confidence* even if their support is less than *minimum*-support, while and $T(S(i)^{p}_{|P|}) - T(S(i)^{p}_{1}) < G_{max}$ and $T(S(i)^{p}_{|P-1|}) - T(S(i)^{p}_{1}) > G_{min}$. In our definition *support* is the frequency count of a sequence $S(i)^{p}_{s,e}$ =$<s_s,s_2,...s_e>$ in *D*, and confidence will be as: *Frequency count of* $S(i)^{p}_{s,n}$.= $<s_s,..., s_n>$ /

Frequency count of $S(i)^{p}_{s,n-1}$ = <s_p,..., s_{n-1}>. As it shows we are looking for *if-then* type of rule *(P ⇒ Q)* in which *if* part refers to $S(i)^{p}_{s,n-1}$ and *then* part refers to $S(i)^{p}_{s,n}$.

3. RELATED WORK

Sequential pattern mining is an important data mining problem with broad applications, including the analyses of customer purchase behavior, Web access patterns, scientific experiments, disease treatments, patient database, natural disasters, DNA sequences, Network data analysis etc. In AI a lot of work has been done for discovering patterns in sequential data [12] [8]. In the database context, where input data is usually much larger, the problem has been studied in a number of recent papers [6, 18] [14] [4]. In [14] event sequences are searched for frequent patterns of events. These patterns have a simple structure (essentially a partial order) whose total span of time is constrained by a window given by the user. The technique of generating candidate patterns from sub-patterns, together with a sliding window method, is shown to provide effective algorithms. In [4] the problem of discovering sequential patterns over large databases of customer transactions is considered. Similarly to [14], the strategy of [4] is starting with simple sub-patterns (subsequences in this case) and incrementally building longer sequence candidates for the discovery process (Apriori Algorithm). Almost all of the previously proposed methods for mining sequential patterns and other time-related frequent patterns are apriori-like, which states the fact that any super-pattern of a non-frequent pattern cannot be frequent. Based on this heuristic, a typical-like method such as GSP [18] adopts a multiple-pass, candidate-generation and test approach in sequential pattern mining.

Han et al proposed a technique to mining sequential data without candidate generation [10]. They introduced frequent pattern tree structure, which is an extended prefix tree structure for storing compressed, crucial information about frequent patterns, and develop an efficient FP-tree-based mining method, FP-growth, for mining the complete set of frequent patterns by pattern fragment growth. The work in [20] also deals with the discovery of sequential patterns. In [20] the considered patterns are in the form of specific regular expressions with a distance metrics as a dissimilarity measure in comparing two sequences. In [21] a scenario is considered where sequential patterns have previously been discovered and an update is subsequently made to the database. Das et al in their work [7] presents a new method for rule discovery from time series data. They slide a window over data and find the class of the subsequence and then find common episodes in represents.

Many researchers have studied the problem of too many rules and discovering interesting rules (e.g., Piatesky-Shapiro & Matheus 1994; Klemetinen et al 1994; Silberschatz & Tuzhilin 1996; Liu & Hsu 1996; Padmanabhan & Tuzhilin 1998) have been proposed to help the user find interesting rules from a large number of discovered rules. The main approaches are either using some interestingness measures to filter out those uninteresting rules or using the user's domain. Cohen et al studied the finding rules with high confidence and low support. They developed a family of algorithms, employing a combination of random sampling and hashing techniques.

There are a few works, which attempt to incorporate the self-similar information in the association rule discovery. Barbara introduced using the fractal dimension to analyze how association rules occur in a dataset [5]. They developed a two tire techniques. First, as a k-itemset is under consideration, and they are scanning the dataset to compute its support, they also roll a window and compute the fractal dimension of the occurrence of this rule as the algorithm goes through the data. Secondly, if this itemset is found to have a lot of support, enough information about the fractal dimension of this rolling window will be kept to be used when processing the k+1 extensions of this itemset in the next iteration of the algorithm. Our work is different with [5] as we address scalability and *interesting rules* rather than using fractal dimension to find those rules.

4. TIEM INVARIANT SEQUENTIAL ASSOCIATION RULES

The notion of time-invariant sequential association rules refers to a group of discovered rules in which a change such as scaling in time constraints implies a new rules with enough *support* or *confidence*. As pattern in smaller scale repeats in larger scale, a new rules may discover by the change in time constraint of discovered rules such as G_{min} and G_{max}. Time-invariance space is broader than self-similarity space. While self-similarity has to be maintained in different scale of a given data, the notion of time-invariance does not need such requirements and may only maintains in one or two scales.

On the other hand, the notion of self-similarity in general helps to understand the concept of time invariance in a continuous time series or discrete database. In the following, we review briefly the notion of self-similarity in continuous domain and we provide a detail definition for time-invariant sequential association rules and self-similar sequential association rules.

4.1 Self Similarity

The mathematical study of self-similar shapes and their relationship to natural shapes was first presented by Benoit Mandelbrot. Self-similar stochastic processes were introduced by Kolmogorov in a theoretical context and brought to the attention of probabilists and statisticians by Mandelbrot and his co-workers and have been used in hydrology, geophysics, biophysics, and biology and communication systems [13].

In general, self-similarity or long range dependence refers to observation of similar patterns when a discrete or continuous time process is scaled in time. The process in larger scale is a copy of itself in smaller time scales. In self-similar signals the key parameter is not the mean or variance, but the degree of self-similarity, defined via the Hurst parameter. The notion of self-similarity is not merely an intuitive description but a precise concept captured by the following rigorous mathematical definition. Let X be wide sense stationary process, that is; a process with constant mean and finite variance and auto correlation function r(k). For each m=1,2,..., let $X^{(m)}$ denotes a new time series obtained by averaging the original series X over non-overlapping blocks of size *m*. That is for each *m*=1,2,... , $X^{(m)}$ is given by $X_k{}^{(m)} = 1/m(X_{km-m+1} + ... + X_{km})$, which $K \geq 1$. Note that for each m, the aggregated time series $X^{(m)}$

defines a wide sense stationary process; let $r^{(m)}$ denote the corresponding auto correlation function. The process X is called exactly H-self similar if for all m>0 it holds

$$X_k = m^{-H} \sum_{i=(k-1)m+1}^{km} X_i$$

By looking at a self-similar sequential data generated through recurrent process, a macro point of view suggests that the overall system behavior is more a trajectory among phases. As a self similar process repeats it self , a pattern will be repeated in different scales. Discrete self-similarity share the same characteristics with continues domain. In the following we define the time-invariant sequential association rules and self-similar sequential association rules.

4.2 Time-Invariant Association Rules

In the following, we provide definition of time invariance and self-similarity for association rules along with some examples.

Time-invariant Sequential Association Rule: A sequential association rule such as P define as a frequent pattern, if it satisfies user defined *support* and *confidence* with Gp_{min} and Gp_{max} (Gp_{min} refers to G_{min} and Gp_{max} refers to G_{max} for pattern P). P is time-invariant if there is a pattern such as Q, with Gq_{min} and Gq_{max}, satisfies *minimum-support* or *minimum-confidence* when Gq_{min} = K_{min} * Gp_{min} and Gq_{max}= K_{max} * Gp_{max} (Gq_{min} refers to G_{min} and Gq_{max} refers to G_{max} for pattern Q). Pattern Q calls interesting if it does not satisfy the *minimum-support* but satisfies the *minimum-confidence*. Note that in most of the cases K_{min} and K_{max} belong to the same order of magnitude and $K_{\min}, K_{\max} \in R$.

Self-Similar Sequential Association Rule: A pattern Q calls self-similar sequential association rules if Q known as time-invariant sequential association rule and if Q maintain such property for different scale of K_{min} and K_{max} in which $K_{\min}, K_{\max} \in R$.

A self-similar data exhibits fractal dimensionalities up to a certain level and has been generated through a recurrent process. The fractal dimension of a time series and self-similarity may validate through well-known algorithms such as introduced in [11].

Example 1: Assume S = <1,X,Y,1,-1,1,0,X,X,X,2,Y,1,2,X,Y,1,X,0,-1,1,X,-1,Y>. Pattern P = <1,[I]*,X,[I]*,Y> with G_{min}=3 and G_{max}=4 will have *support* = 3 and *confidence* = 3/5. If we scale up this pattern with G_{min} =6 and G_{max}=12 the *support* and *confidence* will be 2 and 2 respectively.

Example 2: Table 1 shows an example of Critical Care database. There are two patients in the database with a series of perceptions and actions. No patient has more than one transaction with the same transaction-time. We do not consider quantities of given treatment applied to patient for this stage: each item is a variable representing which treatment was given to the patient or not. A treatment set is a non-empty set of treatments. A sequence is an ordered list of perceptions (signs) and actions (treatments).

For the pattern P=<1,[I]*,X,[I]*,1> with Gp_{min}= 10 and Gp_{max}= 100, *support* is equal to 5 and *confidence is 5/8*. If we set the minimum support equal to 4, P passs the support filter and it is in the result. However, for the pattern R=<1,[I]*,X,[I]*,1> with Gr_{min}= 150 and Gr_{max}= 300 does not pass the support filter, because even though its *confidence* is equal to 1 but its *support* is equal to 2.

The time-invariant sequential association rules provide such facilities to capture R having P with enough support. Interpretation of a sequential association rule plays an important role in rule understanding and rule scaling. For instance pattern P and R can be interpreted as:

> *"if treatment A applied to patient right in 10 time units, good response will be observed in less than 100 time units".*

Similar to P, Pattern R can be interpreted as :

> *" If a treatment has applied to a patient in 150 time unit, patient most probably response in a longer period (300 time units)".*

5. METHOD

The general approach is to find the frequent sequences which satisfy the support level, G_{min} and G_{max} in first step and scale the discovered rules in second step to obtain new sequences which satisfies an acceptable level of *confidence*. In a departure of previous techniques the main contribution of this methods comes from the structure embedded in self-similar data. The self-similarity implies a rule in smaller scale may repeats in larger scale even though if it may does not satisfy the user defined *minimum-support*. is In the following we explain the major steps with a simple pseudo code to capture such rules:

1. Find patterns with support greater than *minimum-support*. This part of the algorithm would be similar to the most of the existing approaches. The main difference is in the frequency count part of a pattern in which we apply the notion of *min-gap* and *max-gap*. Figure 1 shows the pseudo algorithm to find the frequency count of a pattern in a sequence considering *min-gap* and *max-gap*. As it shows the algorithm uses of a dynamic programming like algorithm to capture * factor and gap constraints in the pattern. Similar to an a-priori like algorithm L keeps track of matches in pattern and sequence. When a match occurs, $L(i,j)$ increases when there is at least a match in past items unless if it is a duplicate pattern. F keeps track of time difference which will be checked for G_{min} and G_{max}. T keep the exact time of a perceptions and actions. Note that the sequence scanned only once and the order of the a-priori like algorithm has not changed.
2. If there is not enough frequent pattern found in the data, change the G_{max} to a greater value. This increases the possibility of observing frequent pattern in a sequence as * play an important role in frequency count.
3. Since data is self-similar or has shown self-similarity up to a certain degree, for all frequent patterns, scales up the rule by scaling up the G_{max} and G_{min}.
4. Scan D from the beginning and compute the frequency count of the new rule. Store new rules if their confidence is greater than user *minimum-confidence*. These rules are essentially interesting cause they are not intuitive, but they could happen only a few times for a perception-action data. As the number of occurrences of these rules is relatively low they never recognized in a-priori like algorithms, which are *support-based* algorithms.

Figure 1: Pattern Matching Considering G_{min} and G_{max}

```
FindFrequentPattern(Pattern,Sequence,$G_{max}$,$G_{min}$);
initialization
Loop in Sequence i
 Loop in Pattern j
   IF Sequence(i) == Pattern(j)
     IF  i==1 | j==1
       L(i,j)=1;   keep track record of  matches
       F(i,j)= 0;  get the time difference
       T(i,j) = i-1; the exact time
     Else IF L(i-1,j-1) ~= 0
       if window condition satisfies
         L(i,j)=L(i-1,j-1) + 1;
         F(i,j)= temp;T(i,j) = i-1;
     Else IF find all non zero members in previous column
            set index to most recent none zero
        IF it's not duplicate
                IF it satisfies window condition
          L(i,j) = L(index,j-1) + 1;
          F(i,j) = (i-1-T(index,j-1))+F(index,j-1);
          T(i,j) = i-1;
       Else   no match
         L(i,j)=1; F(i,j)=0; T(i,j)=i-1;
       IF  L(i,j)== length of Pattern
         count=count+1;
      ELSE
       L(i,j)=0; F(i,j)=0; T(i,j)=i-1;
```

5.1 Scale Factor

Scale factors (K_{min} and K_{max}) has employed to provide new constraints as:

$$T(S(i)_{|P|}^{P}) - T(S(i)_{1}^{P}) < K_{max} * G_{max}$$

$$T(S(i)_{|P-1|}^{P}) - T(S(i)_{1}^{P}) > K_{min} * G_{min}.$$

The scale factor basically is very subjective and has a strong bound with the domain knowledge, user input and user preferences. For a Medical database a scale factor is from *minute, 15 minutes* and *an hour* up to *8 hours, 12 hours, 24 hours, 96 hours*. For a Network data base scale factor is from *10 minutes, 20 minute, 30 minutes* and *an hour*, up to *a week*, and *a month*.

The scale factor may consider as hidden information. The lack of such knowledge is similar to hidden information such as number of clusters in a clustering problem or number of state in a Markov modeling problem. However, similar to those problems scaling factor has strong roots in the nature of the problem itself and it can provide either by user, using a heuristic or through a search process.

5.2 Analysis

There are two major issues in knowledge discovery loop which has to be considered. These issues are Time and Space.

Number of scan over the time series: a-priori like algorithm scans databases P time that is equal to number of *L-length* patterns satisfy the *minimum-support*. If we show the discovered rules in step 1 as R and the length R with L_R=|R|, we would scale up each rule C time depends on the fractal dimension of the data or as much as user specifies. In this case we scan database L_R . C in the worst case.

Space needed: If all scaled up association rules would satisfy the *minimum-confidence*, then in the worst case with an average length of L_A for all discovered rules we need L_A.|R|.C more space comparing to a-priori algorithm which is negligible comparing to the whole dataset.

6. RESULT

The notion of time-invariant sequential patterns has been investigated in the context of Critical Care domain. The database is a collection of two different sets of patients from King Drew Medical Center (for patient going to Intensive Care Unite mainly because of accident, gun shots and/or injuries) and Harbor UCLA Hospital (mainly for senior citizens). Our database has collected during past 15 years. We applied our test only on the selected adequately monitored patients. In addition, data has been considered only after the first surgery as the data during the surgery is not valid due to the high hemorrhage of the patient.

Our implementation is in *MATLAB* programming language and has been tested on Pentium III processor with 384 MB RAM. This study is a follow up on work by Adibi et al [1] in which a complete decision support system designed in Lisp language under Apple/McIntosh platform [15]. We do not address the feature selection problem here and we follow the guideline provided by [15].

6.1 Critical Care Domain

Time-invariant or self-similar sequential association rules play an important role in the context of Critical Care since time is a crucial factor in Critical Care . For instance, admission time, visit time, surgery time, treatment time etc. are examples of association of time and patient care in Critical Care unit.

Our approach is based on the well studied concept that irrespective of multiply of superficial clinical manifestations, the patient dies of physiological alternations that can be identified, and prevented. Shoemaker et al showed that the temporal patterns of postoperative survivors were found to be different from those non-survivors despite the wide variety of illness and operational. [15, 16]

The survivor and non-survivor patterns and their importance of oxygen transport pattern were confirmed by independent investigations [15, 16]. In addition, it has been showed that the increased delivered oxygen (*Do2*) and consumed oxygen (*Vo2*) patterns of early postoperative survivors are clearly separate from the relatively normal values of non-survivors [15, 16]. Similarly, in other etiologic types of shock the survivor patterns are higher than those of the non-survivors at comparable time periods. However the main question of such protocol is to find under which circumstances a patient states moves form survivor to

Table 3: Discovered Rules in Critical Care Database

Survivors	Pattern 1	Pattern2	Pattern 3
Support	188	381	358
Confidence	.20	.41	.39

Non-Survivors	Pattern 1	Pattern2	Pattern 3
Support	223	389	365
Confidence	.23	.40	.37

non/survivor and to find which patterns has been repeated in survivor and non-survivor patients.

The physiology of postoperative and years of study in this field [16] shows distinguishes property in *Do2/Vo2* diagram in first 8 hours. There are two major patterns in survivors or non-survivors plot. After the first 8 hours it would be a significant difference in survivors and non-survivors pattern.

6.2 Experimental Result

For the purpose of the validating of out method, we conducted a multi-step experiment on our current database as the following:

1. First we pick all adequately monitored patients from survivors and non-survivors groups.
2. We applied the a-priori like algorithm on this set. We set G_{min} to 1 *hour* and G_{max} to *8 hours* after surgery.
3. We scaled up the discovered rules by K equal to *24* and *96* hours after surgery for G_{max} .

The result is interesting and shows rules with a low *support* and high *confidence*, which did not come up in the first step, will be discovered. The main idea is to find the effectiveness of treatment in increasing the probability of patient as being as a survivor at the end of the procedure by measuring the ratio of Vo2/Do2. We show the trend of this probability with 1: increases, -1: decreased, and 0: unchanged.

The following are the list of some interesting patterns were for we found for $G_{max} = 8$ *h*ours ad $G_{min} = 1$ hour :

Pattern 1. $P= <[I]*,A,[I]*,1>$ indicate patient response to a given treatment

Pattern 2. $P=<[I]*,A,[I]*,0>$ indicate no change in patient condition after giving a treatment

Pattern 3. $P=<[I]*,A,[I]*,-1>$ indicate no response from the patient to a given treatment which A is the treatment.

We scale up such rule and apply to survivors and non-survivors for K = 3 and 12. the observation was that if the treatment applied to patient was in the goal of bringing up the *Do2/Vo2* it would save patient live more probably. When any treatment, which has the capability to increase the level of delivered Oxygen to the patient, is applied late the response of the patient also has been late and sometimes also is too late to recover.

The discovered rule might not consider as hard-to-find rules or hidden rules. There are a huge set of discovered rules. However we only discuss those which are easier to interpret and they are interesting for physicians and health care providers. Even thought these result are in early stage but they are encouraging and we would like to investigate more undiscovered rules and apply to other large databases.

7. CONCLUSION AND FUTURE WORK

Despite the broad range of research on sequential association rules, they could not easily discover rules with low *support* and high *confidence.* We refer to this series of rules as *interesting rules* that are important in a relatively broad range of application in science, technology and medicine. In this paper we provide a fairly simple but powerful formalism to extend a pool of discovered rules to capture interesting rules for a specific databases with unique characteristics. The information and complexity embedded in these collections are hierarchical, they are self-similar, contain self-similar structures and have been generated through recurrent processes. We introduced time-invariant sequential association rules as those rules which if extend in time dimension explore more knowledge form data. Since time-invariant sequential association rules are capable to discover a relatively hard-to-find association rules, they may extent to all database which shows partially self-similarity.

This research is in early stage. As future work we would like to continue our findings in Critical Care domain and extend this research to multi dimensional sequential association rules. In addition we are in the process of discovering rules in strongly self-similar time series such as synthetic data or network databases. The nest step for Critical Care domain is considering the dosage of in discovery loop. In addition we would like to extend this work when the model shows self-similar structure only in a limited range of structure scale.

8. ACKNOWLEDGEMENT

This work was partially supported by the National Science Foundation under grant: NSF-IDM 9529615. We would like to thank Dr. William C. Shoemaker for his tremendous effort on data analysis, rule interpretation and his advise on Critical Care domain.

9. REFERENCE

[1] Adibi, J., Patil, R. S., and Shoemaker W. C. *A Perception-Action model for Critical Care.* in *American Medical Informatics Association.* (1997). Nashville, Tennessee.

[2] Adibi, J., Shen, W-M. *Self Similar Layered Hidden Markov Model.* in *5th European Conference on Principles and Practice of Knowledge Discovery in Databases (PKDD'01).* (2001). Freiburg, Germany.

[3] Agrawal, R., Lin, K. I., Sawheny, H. R., and Shim, K. *Fast similarity search in the presence of noise, scaling and translation in time series databases.* in *VLDB.* (1995). Zurich, Switzerland.

[4] Agrawal, R., Srikant, R. *Mining sequential patterns.* in *the Int'l Conference on Data Engineering (ICDE).* (1995). Taipei, Taiwan.

[5] Barbara, D. *Chaotic Mining: Knowledge discovery using the fractal dimension.* in *ACM SIGMOD Workshop on Research Issues in Data Mining and Knowledge Discovery (DMKD).* (1999). Philadelphia, USA,.

[6] Cohen, E., Datar, D., Fujiwara, S., Gionis, A., Indyk, P., Motwani, R., Ullman, J. and Yang, C. *Finding interesting associations without support pruning.* in *16th Annual IEEE Conference on Data Engineering (ICDE).* (2000).

[7] Das, G., Lin, K., Mannila, H., Renganathan, G. and Smyth, P. *Rule discovery for time series.* in *Int'l Conference on Knowledge Discovery in Databases and Data Mining (KDD-95).* (1995). New York City, New York.

[8] Dietterich, T.G., Michalski, R. S., *Discovering patterns in sequences of events.* Artificial Intelligence, (1985). **25**: p. 187-232.

[9] Faloutsos, C., Seeger, B., Traina, A. and Traina Jr., C. *Spatial Join selectivity using power law.* in *SIGMOD.* (2000). Dallas, TX.

[10] Han, J., Pei, J. and Yin, Y. *Mining frequent patterns without candidate generation.* in *ACM-SIGMOD Int. Conf. on Management of Data (SIGMOD'00).* (2000). Dallas, TX.

[11] Higuchi, T., *Approach to an irregular time series on the basis of the fractal theory.* Physica D, (1998). **31**.

[12] Laird, P. *Identifying and using patterns in sequential data.* in *Algorithmic Learning theory, 4th International Workshop.* (1993): Springer-Verlag.

[13] Mandelbrot, B., Van Ness, J. W., *Brownian motion fractional noises and applications.* SIAM review, (1968). **422**(437).

[14] Mannila, H., Toivonen, H., and Verkamo, A. I. *Discovering generalized episodes using minimal occurrences.* in *Second International Conference on Knowledge Discovery and Data Mining.* (1996). Portland, Oregon.

[15] Patil, R.S., Adibi, J., and Shoemaker W. C., *Application of an Artificial Intelligence program to therapy of high risk surgical patients.* New Horizons The Science and Practice of Acute Medicine, (1996). **4**(4): p. 541-550.

[16] Shoemaker, W.C.S., Patil, R. S., Adibi, J. et al., *Early physiologic patterns in acute illness and accidents: Toward a concept of circulatory dysfunction and shock based on invasive and noninvasive homodynamic monitoring.* New Horizons The Science and Practice of Acute Medicine, (1996). **4**(4): p. 395-412.

[17] Silverstein, C., Brin, S., Motwani, R., and Ullman, J.D. *Scalable Techniques for Mining Causal Structures. In Proceedings of the.* in *24th International Conference onVery Large Data Bases.* (1998).

[18] Srikant, R., Agrawal, R. *Mining sequential patterns: Generalizations and performance improvements.* in *Fifth Int'l Conference on Extending Database Technology (EDBT).* (1996). Avignon, France.

[19] Traina, C., Traina, A., Wu, L., and Faloutsos, C. *Fast feature selection using the fractal dimension.* in *XV Brazilian Symposium on Databases (SBBD).* (2000). Paraiba, Brazil.

[20] Tsong-Li Wang, J., Chim, G., Marr, T.G., Shapiro, B. A., Shasha, D., and Zhang, K. *Combinatorial pattern discovery for scientific data: Some preliminary results.* in *SIGMOD.* (1994): ACM Press.

[21] Wang, T., Tan, J. *Incremental discovery of sequential patterns.* in *Workshop on Research Issues on Data Mining and Knowledge Discovery, in cooperation with ACM-SIGMOD.* (1996). Montreal, Canada.

Modeling Sparse Engine Test Data Using Genetic Programming

Tina Yu
Chevron Information Technology Company
6001 Bollinger Canyon Road
San Ramon, CA 94583
U. S. A.
001-925-842-2393
tiyu@chevron.com

Jim Rutherford
Chevron Oronite Company LLC
100 Chevron Way
Richmond, CA 94802
U. S. A.
001-510-242-3410
jaru@chevron.com

ABSTRACT

We demonstrate the generation of an engine test model using Genetic Programming. In particular, a two-phase modeling process is proposed to handle the high-dimensionality and sparseness natures of the engine test data. The resulting model gives high accuracy prediction on training data. It is also very good in predicting low range data values. However, at least partly due to limitations of the data set, its accuracy on validation data and high range data values is not satisfactory. Moreover, the subject experts could not interpret its real-world meaning. We hope the results of this study can benefit other engine oil modeling applications.

Keywords

Data Modeling; Genetic Programming; Sparse Data; High Dimensionality; Virtual Testing.

1. INTRODUCTION

Laboratory engine tests are among the tools used to measure engine oil performance. These tests are specified in various engine oil performance categories for licensing and certification [3][4][12]. Lubricant additive companies and engine testing laboratories implement and exercise these tests to produce high-quality engine oil.

One of the engine tests used is Sequence IIIE. Early in the year 2000, capability to run this test had nearly been eliminated due to engine parts becoming unavailable. In response to this change, the American Society for Testing and Materials (ASTM) Sequence II/III Surveillance Panel formed the Virtual Test Task Force (VTTF) in May of 2000. The mission of VTTF was to investigate and develop a process, if appropriate, for the use of mathematical models based on IIIE data as a substitute for the Sequence IIIE test.

A virtual engine test protocol was subsequently devised and reported back to the Panel after four months of investigation. However, the proposed process did not receive enough support to be implemented. We believe that it is neither technical nor practical issues that hinder the implementation. Instead, it is the lack of familiarity and comfort with the proposed procedures that prevents the adoption of virtual testing [23].

In this work, we demonstrate how an engine test model can be created using Genetic Programming (GP) [14]. It is hoped that through understanding the data modeling process, the related organizations will become more comfortable with the concept of virtual engine testing. Moreover, we hope other engine oil modeling applications can benefit from this study.

The paper is organized as follows. Section 2 explains the Sequence IIIE engine test data. Section 3 presents GP algorithm as a data-modeling tool. In Section 4, experimental setup is given and in Section 5, the experimental results are presented. Section 6 gives our analysis and Section 7 discusses the results of the study. Section 8 reviews related work and Section 9 contains the conclusions.

2. SEQUENCE IIIE ENGINE TEST DATA

The test has been running for over 10 years. As a result, we have a relatively large data set. However, many of the data have missing information. For example, many potential predictors such as base oil characteristics were not recorded. We made improvement on 172 data records, which are used in this study to generate an engine test model.

There are nine passing criteria for the Sequence IIIE engine test [4]. The criteria are *percent viscosity increase, average piston varnish, average camshaft plus lifter wear, maximum camshaft plus lifter wear, average engine sludge, oil ring land deposits, oil consumption, oil related stuck rings*, and *stuck lifters*. A complete engine test system is a suite of nine models; each model predicts one of the nine passing criteria. In this work, we focus on the viscosity increase model. The methodology can be applied to generate other models.

Besides the test results (for the nine passing criteria), each test record contains information about the ingredients of the tested engine oil. For example, viscosity index improver (VII) and dispersants are common engine oil additives. Due to the diversity of the additives and complex naming conventions, the number of additive variables is large (109). Moreover, it is common for an additive to be present in very few of the data records due to the experimental nature of oil formulation. As a result, the data set is very sparse.

Figure 1 shows that 28% of the 109 additive variables appear only in one test record within the entire data set. More than 50% of the 109 variables appear in less than 5 test records. The combination of high-dimensionality and sparseness has made the engine test data difficult for most data modeling tools.

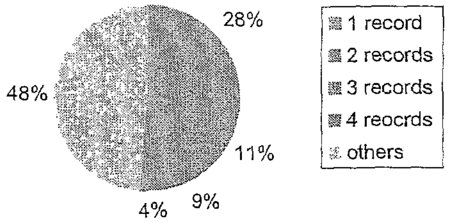


Figure 1: Variables in the data set.

2.1 Aggression and Distribution

Data aggregation and distribution are mechanisms to organize data sets. In this study, we group the additive information into "families" to reduce the size of variables and to increase the density of the data set.

Initially, the expertise of engine oil formulators was used to rearrange and collapse variables in the data. We group the list of 109 additives into 13 families of similar additives. In some cases, additive concentration was simply the sum of concentrations of the additives in the family. In other cases, equivalency relationships based on known or suspected mechanisms were applied. For example, equivalent antioxidancy was derived for the various antioxidants based on chemical functionality. The number of additives in each family varies, ranging from 2 to 25.

After the family grouping is defined, each family is represented with two columns in the data set: one column contains the additive name and the other gives the additive amount used. Table 1 shows the aggregated format for VII additives. If an additive family is not present in a test record, the additive-name is "none" and the additive-amount is 0.

Table 1: Aggregated formats for VII additives.

...	...	VII-name	VII-amount	...
...	...	vii-name-1	0.256	...
...	...	none	0.0	...
...	...	vii-name-2	21.3	...

With this aggregation method, the 109 additives are reduced to 26 variables in the data set. Adding other testing related information, such as end of test date, viscosity grade, and base oil characteristics, the total number of variables is 39. At the end of this aggression process, not only the number of variables is reduced, the density of the data set is also increased.

We used this data set for SGI MineSet [16] to generate a regression tree using its default setup:

- The software performs the splitting of training and testing data in a random manner.
- No cross-validation is performed.
- The software uses a normalized mutual information as the splitting criteria for tree nodes.
- The software uses a confidence-based algorithm to perform tree pruning.

The following model is generated in one run (note that the status window shows the number of training data is 115 while the number of testing data is 57):

```
Viscosity Increase =
        If (saturates <= 98.18) then 118.478
               else if detergent <= 13.473
               then  170.333
                      else 5242.8
```

This result is not satisfactory, as its accuracy (mean absolute error 1007.9) is not good enough to be a useful engine test. We believe the inherent multicollinearity of chemical additives is a challenge to most modeling tools, such as neural networks, support vector machines and linear regression.

As the first attempt to explore the possibility of modeling such a data set using GP, we applied Discipulus software [9] to generate a mathematical expression model (see Section 3 for examples). This approach requires two phases because the model representation in this GP software does not support categorical values (e.g. VII-name).

In the first phase, the 13 additive-name columns (categorical variables) are removed from the data set. The number of the variables is reduced to 26. The purpose of this modeling phase is *features selection*. In the second phase, each of the selected additive-amount variables is expanded with its associated additive name (column distribution). Table 2 shows the distributed format for VII additives (This is the original format before aggregation).

Table 2: Distributed formats for VII additives.

...	...	VII-name-1	VII-name-2	...
...	...	0.256	0.0	...
...	...	0.0	0.0	...
...	...	0.0	21.3	...

In the following sections, we will present the work using Discipulus and the two-phase modeling process to generate an engine test model.

3. GENETIC PROGRAMMING

GP is a machine learning algorithm that is suitable for data modeling [5]. Figure 2 depicts the GP algorithm cycle:

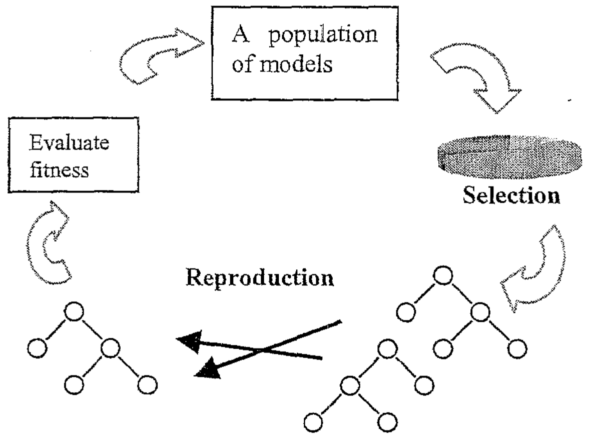


Figure 2: GP algorithm cycle.

Initially, a population of models is randomly created. Based on their fitness, better models are selected for reproduction. Using alternation operations, such as crossover and mutation, new offspring models are generated for fitness evaluation. This process of selection, alternation and fitness evaluation continues until a satisfactory model is generated.

Various representations, selection and alternation schemes have been proposed to suit different applications [25][26]. The Discipulus GP software uses a linear representation to generate mathematical expressions. The following is an example model:

```
Viscosity Increase =

     3.4*detergent+ 3*saturates²

     - aromatics/visindex - 9
```

4. EXPERIMENTS

The 172 data records come from two different engine test laboratories. We used data from one laboratory (104) for training and the other (68) for validation.

In Discipulus, training data is used to evaluate the fitness of the evolved models. This is the fitness that selection for reproduction is based on. In contrast, validation data do not participate in the model generation process. It serves as an unseen data set to give an indication of the robustness of a model. Validation fitness is the selection criterion for the final model, in order to avoid overfitting.

A dynamic training subset selection mechanism [10] is implemented in Discipulus. The subset selection criteria include difficulty, age and randomness. We considered using this feature but decided not to due to the small size of the data set. We believe different results would have been produced if this feature were applied.

Table 3 summarizes the parameters used to conduct the experiments.

Table 3: GP parameters.

Objective	Generate a model that predicts viscosity increase values.
Functions	addition; subtraction; multiplication; division; abs; sqrt; data transformation
Terminals	1st phase: 26 variables 2nd phase: variables selected from 1st phase Constants: 0, 0.5, 1.
Fitness	Linear absolute error
Selection	Tournament (4 candidates/2 winners)
Pop size	100,000
Max Gen	9,000,000
Max Length	256
Genetic Operators	50% crossover (Homologous 95%), 95% mutation (Block mutation rate 30%, Instruction mutation rate 30%, Instruction mutation rate 40%)

5. RESULTS

We made 10 runs and the final model is the one with the best validation fitness:

```
Viscosity Increase = F + 2 * abs (G),
```

where F and G are equations, defining additive usage relating to different oil characteristics. The interpretation of its real-world meaning is not clear (see Section 7).

We used five measurements to evaluate the generated models (other measurements such as uncertainty will be included in the future work). Table 4 summarizes the results of the final model.

Table 4: Experimental results of the final model.

	Training Data	Validation Data
Mean Error	37.89	76.89
Median Error	18.14	48.81
Worst Case Error	282.86	452.03
Correlation	0.98	0.71
Coefficient of Variation (R^2)	0.96	0.50

The three accuracy measurements (mean, median and worst case errors) are calculated on data records whose target viscosity-increase values are less than 1000. This means that three records in the training data and four records in the validation data are excluded from the calculation. This decision is based on the fact that 375 is the maximum allowable viscosity-increase to pass the engine test (see Table 5). Beyond this threshold, as long as the model gives a > 375 prediction, it meets the business needs. Indeed, the GP model gives a high enough value for each of these seven cases to indicate that they fail the test.

The relationship measurements (correlation and R^2) on training data are very good (0.98 and 0.96). However, those on validation data are not as impressive (0.71 and 0.50). Similarly, the accuracy measurements (mean, median and worst case errors) on training data are far superior to those on validation data. Section 7 will provide some possible explanations of such discrepancies.

6. ANALYSIS

Depending on the performance category that the engine oil is tested for, different viscosity increase limits are allowed (see Table 5). For example, the maximum percent viscosity increase value for API CH-4 category is 200. Any value within this threshold is acceptable. The same applies to the other two thresholds (100 and 375).

Table 5: Viscosity increase thresholds vs. test category.

Category	Viscosity Increase (%)
API SG, SH, SJ; ILSAC GF-1 GF-2	375 maximum
API CH-4, ACEA A2-96	200 maximum
ACEA A1-98, A3-98	100 maximum

For the purpose of issuing licenses, what is required of a testing system is its ability to predict whether the performance of the tested engine oil is within the required threshold or not. The actual prediction value is not as important. Based on this merit, the engine test model is performing a classification task; it classifies the tested engine oil to be in one of the following 4 viscosity-increase ranges:

- < 100
- between 100 and 200
- between 200 and 375
- > 375

We analyze the accuracy of the GP model in classifying the engine test data using confusion matrices.

In Table 6 and 7, each row represents the actual values while the column gives the predicted value. As shown, the model is very good at predicting < 100 range. Within the training set, there are 69 such kind of records; the model correctly predicted 66 of them (96% accuracy rate). The accuracy rate on validation data is 91% for this range. Between the range of 100 and 200, the performance drops (24% on training data and 0% on validation data). The model made no correct prediction on 200 to 375 range values. For data value > 375, the accuracy is 100% on training data and 40% on validation data. The overall accuracy is 73% on training data and 50% on validation data.

Table 6: Confusion matrix analysis on training data.

(a)

A P	<100	100-200	200-375	>375	Total
<100	66	3	0		69
100-200	20	7	2	0	29
200-375	3	0	0	0	3
>375	0	0	0	3	3
Total	89	10	2	3	104

(b)

A P	<100	100-200	200-375	>375	Total
<100	96%	4%	0%	0%	100%
100-200	69%	24%	7%	0%	100%
200-375	100%	0%	0%	0%	100%
>375	0%	0%	0%	100%	100%
Total					73%

Table 7: Confusion matrix analysis on validation data.

(a)

A P	<100	100-200	200-375	>375	Total
<100	32	2	1	0	35
100-200	18	0	0	0	18
200-375	10	0	0	0	10
>375	3	0	0	2	5
Total	63	2	1	2	68

(b)

A P	<100	100-200	200-375	>375	Total
<100	91%	6%	3%	0%	100%
100-200	100%	0%	0%	0%	100%
200-375	100%	0%	0%	0%	100%
>375	60%	0%	0%	40%	100%
Total					50%

The 0% accuracy rate on data range values between 200 and 375 is the result of small number (3) of training data. As a data-driven modeling method, GP is less likely to generate a good model without enough training data.

7. DISCUSSION

After presenting the model to subject experts, some concerns were raised. First, the accuracy on validation data is much lower than that on training data. We investigated the characteristics of training and validation data and found there are many differences.

For example, eight validation data have large quantities (e.g., 1074 or 1236) of equivalent antioxidancy that produce low viscosity-increase values (<200). In contrast, this equivalent antioxidancy is of much smaller quantities (e.g., 267, 537, etc.) in the training data. Another example is a frequently used dispersant in training data is hardly used in validation data. Furthermore, validation data used ZNDTP A much more often than ZNDTP B while training data is the other way around. Such discrepancies have made it difficult for GP to generate a common model that works well for both data sets.

These differences, although confounded with the laboratories, do not seem to be caused by laboratory differences, according to a subject expert. They are probably artifacts of the shifting concentration of testing between the laboratories, while there were concurrent changes of industry testing severity and the change of formulating strategies. This information suggests that we should consider the whole data set as one trend of engine test records. Instead of splitting the data based on laboratory association, it might be more appropriate to split them based on other criterion, such as the viscosity increase data range.

Another suggested method to increase the generality of the model is to use a predictor ensemble. With this approach, each sub-model in the ensemble is trained differently, e.g. by using different partition of the data set or different GP parameters etc. As a result, each sub-model would give a different prediction for the same inputs. The final output of the ensemble is the weighted average of the outputs by all sub-models. Numerous researchers have shown empirically that such ensembles generalize well [6][21].

The second concern that subject experts raised is with model fit on training data. The three extreme high value data (1152, 13519 and 19393) are very influential to the calculation of the relationship measurements. In Figure 3, all data except these three extreme high value data are clustered at the lower left corner. The trend line gives high correlation between the actual and the predicated values.

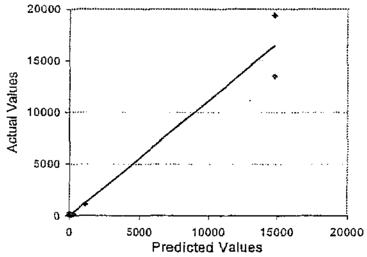


Figure 3: Model fit on training data.

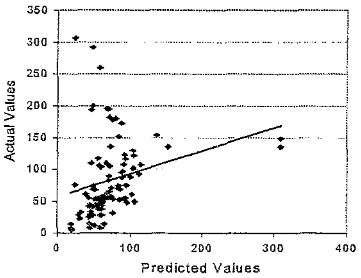


Figure 4: Model fit on training data excluding the three extreme high value data.

However, within the cluster, the correlation between actual and predicted values is not good (see Figure 4). This phenomenon highlights a common dilemma when modeling data with a very wide range of values:

- High value data points are necessary to train a model to be able to predict high range data values;
- However, these high value data points also bias leaning to compromise low range value data.

There are a couple of known methods to work with data set with a wide range of values:

- Convert the data values into logarithm values.
- Customize the fitness function to give proper bias (weight) on both high and low value data. For example, the data with target viscosity increase value greater than 375 can be evaluated with a different standard: when a model gives a prediction greater than 375, the error is 0 on this data point.

Finally, the subject experts also concern with the interpretation of the model. It is hard to attribute real world meaning to terms and operators such as absolute value. Maybe a different representation, one without absolute value operator, is more appropriate.

"The models were not adequate," said one subject expert. "The data should take most of the blame but I also have doubts that GP is an appropriate tool. Performance with the validation set was not good. There were also problems with the model fit to the training data. I don't think this would comfort those people who aren't already comfortable with modeling."

8. RELATED WORK

Using mathematical models for engine testing has been implemented in various applications. For example, Rutherford, Schip and Duteurtre used statistically designed experiments to develop predictions of engine test results from engine oil formulation [22]. Similarly, automotive industry uses mathematical models to predict airflow dynamics instead of wind tunnel testing, or to predict crash performance [19].

U.S. Governments have also adopted the use of mathematical models for testing. The United States Environmental Protection Agency and the California Air Resources Board allow fuel producers to demonstrate clean fuel performance through the use of mathematical models derived from emission test databases [7][8].

In the Machine Learning community, feature selection has long been an active research topic [1]. One approach is using heuristic search algorithms. For example, a rough sets-based algorithm [17] and a Chi2 algorithm [15] were designed to find the relevant features within a larger set of attributes.

Another approach is using decision trees algorithms, such as C4.5 [20]. One result based on the study of Boolean functions indicates that the algorithm is not suitable for filtering irrelevant features [2]. A similar feature selection tool in MineSet is "Column Importance". This algorithm is based on Bayes's theorem; i.e. it assumes the independence of variables. This tool is not appropriate for data sets where interdependency of variables is abundant, such as the Sequence IIIE engine test data.

Genetic Algorithms (GAs) have also been used to perform feature selection in various applications. For example, Yang and Honavar applied a GA to select features from medical data sets [24]. Another work is by Guerra-Salcedo, Chen, Whitley and Smith, who used hybrid GA-based strategies to filter relevant features in 3 different kinds of data set: a satellite, a DNA and a Cloud data sets [11].

Opitz also proposed a genetic ensemble feature selection algorithm (GEFS) to select a set of feature subsets for ensemble [18]. He demonstrated that this approach produces better ensembles on average than that produced by Bagging and Boosting.

9. CONCLUSIONS

Data modeling for testing is not a new concept. Various statistical approaches and machine learning algorithms have been applied to create models from data to perform testing tasks. We demonstrated the data modeling process using GP with data aggregation and distribution. This approach has generated an engine test model that can predict the viscosity increase of engine oil.

The generated model, however, has not received much support from subject experts due to the following reasons:

- Its accuracy on validation data and high range data values is not satisfactory;
- The model fit on training data is biased;
- The representation is not easy to interpret.

We hope to acquire more quality data to improve the accuracy of the model. Meanwhile, methods to adjust GP learning bias will be developed. We are also considering different model representation to better suit the applications.

10. ACKNOWLEDGEMENTS

We would like to thank Wolfgang Banzhaf and the reviewers for their comments and suggestions. We also thank Ileana Krumme for her support in writing up this work.

11. REFERENCES

[1] Aha, D. W. and Bankert, R. L. A comparative evaluation of sequential feature selection algorithms. In *Proceedings of the Fifth International Workshop on Artificial Intelligence and Statistics*, 1995. Springer-Verlag, NY. Pages 1-7.

[2] Almuallim, H. and Dietterich, T. G. Learning Boolean concepts in the presence of many irrelevant features. *Artificial Intelligence*, 69 (1-2), November 1994. Pages 279-305.

[3] API. *American Petroleum Institute Engine Oil Licensing and Certification System*. 1999.

[4] ASTM. *American Society for Testing and Materials D4485-99b*. Standard Specification for Performance of Engine Oils. 1999.

[5] Banzhaf, W., Nordin, P., Keller, R. and Francone, F. *Genetic Programming: An Introduction*. Morgan Kaufmann Publishers, Inc. San Francisco, CA. 1998.

[6] Breiman, L. Bagging predictors. *Machine Learning* 24 (2). 1996. Pages 123-140.

[7] CARB. *California Procedure for Evaluating Alternative Specifications for Phase 2 Reformulated Gasoline Using the California Predictive Model*. California Air Resources Board, adopted April 20, 1995 and last amended December 11, 1999, Sacramento, California.

[8] CFR 40. *Title 40 of the Code of Federal Regulations*, Part 80, Section 80.45.

[9] Discipulus. Register Machine Learning Technologies, Inc. Littleton, CO. 1998.

[10] Gathercole, C. and Ross, P. Dynamic training subset selection for supervised learning in genetic programming. In *Parallel Problem Solving from Nature III*. 1994. LNCS Vol. 866. Pages 312-321.

[11] Guerra-Salcedo, C., Chen, S., Whitely, D. and Smith, S. Fast and accurate feature selection using hybrid genetic strategies. In *Proceedings of 1999 Congress on Evolutionary Computation*. IEEE. Pages 177-184.

[12] JASO. *Japan-America Society of Oregon Engine Oil Standards*. 2000.

[13] Kira, K. and Rendell, L. A. The feature selection problem: traditional methods and a new algorithm. In *Proceedings of the Ninth National Conference on Artificial Intelligence*, 1992. AAAI/MIT Press, Pages 129-134.

[14] Koza, J. *Genetic Programming: On the Programming of Computers by Means of Natural Selection*. MIT Press. Cambridge, MA. 1992.

[15] Liu, H. and Setiono, R. Chi2: feature selection and discretization of numeric attributes. In *Proceedings of the 7th IEEE International Conference on Tools with Artificial Intelligence*, 1995. Pages 338-391.

[16] MineSet. Silicon Graphics, Inc. Version 3.0. Mountain View, CA. 1999.

[17] Modrzejewski, M. Feature selection using rough sets theory. In *Proceedings of the European Conference on Machine Learning*, 1993, Pages 213-226.

[18] Opitz, D. Feature selection for ensembles. In *Proceedings of the Sixteenth National Conference on Artificial Intelligence*, 1999, AAAI/MIT Press, Pages 379-384.

[19] Pescovitz, D. Monsters in a box. *WIRED*, December 2000, pages 340-342.

[20] Quinlan, J. R. *C4.5: Programs for Machine Learning*. Morgan Kaufmann, San Mateo, CA. 1993.

[21] Quinlan, J. R. Bagging, boosting, and c4.5. In *Proceedings of the Thirteenth National Conference on Artificial Intelligence*, 1996, AAAI/MIT Press, Pages 725-730.

[22] Rutherford, J. A. Van't Schip, C. J., and Duteurtre, Ph. Experience with statistically designed experiments in the VW 1431 test. In *Proceedings of the Third International Symposium on the Performance Evaluation of Automotive Fuels and Lubricants*. 1989.

[23] Rutherford, J. Some statistical, technical, and practical issues in virtual engine testing. *Society of Automotive Engineers Technical Paper Series*, No. 2001-01-1906. 2001.

[24] Yang, J. and Honavar, V. Feature subset selection using a genetic algorithm. *In Genetic Programming 1997: Proceedings of the Second Annual Conference*. MIT Press. Pages 380-385.

[25] Yu, T. and Miller, J. Neutrality and the evolvability of Boolean function landscape. In *Proceedings of the 4th European Conference in Genetic Programming*. 2001. LNCS 2083, Springer-Verlag. Pages 204-217.

[26] Yu, T. Structure abstraction and genetic programming. In *Proceedings of 1999 Congress on Evolutionary Computation*. IEEE. Pages 652-659.

Discovering Corrosion Relationships in Eddy Current Non-destructive Test Data

Donald E. Brown
Department Chair and Professor
Department of Systems and Information Engineering
University of Virginia
Charlottesville, VA 22903
804-924-5393
brown@virginia.edu

John R. Brence
Instructor
Department of Systems Engineering
United States Military Academy
West Point, NY 10996
845-938-2746
fj7672@usma.edu

ABSTRACT

Quicker, more effective methods of corrosion prediction and classification will help ensure a safe and operational transportation system for both civilian and military sectors. This is especially critical now as transportation providers attempt to meet the increased expense of repairing aging aircraft with smaller budgets. These budget constraints make it imperative to find corrosion and to correctly determine the appropriate time to replace corroded parts. If the part is replaced too soon, the result is wasted resources. However, if the part is not replaced soon enough, it could cause a catastrophic accident. The discovery of models that limit the possibility of a costly accident while optimizing resource utilization would allow transportation providers to efficiently focus their maintenance efforts. While our concern in this study was with aircraft, the results will also be useful to other transportation providers. This paper describes the discovery and comparison of empirical models to predict corrosion damage from non-destructive test (NDT) data. The NDT data derive from eddy current (EC) scans of the United States Air Force's (USAF) KC-135 aircraft. While we might suspect a link between NDT results and corrosion, up until now this link has not been formally established. Instead, the NDT data have been converted into false color images that are analyzed visually by maintenance operators. The models we discovered are quite complex and suggest that in data mining we can sometimes more effectively handle noisy data through more complex models rather than simpler ones. Our results also show that while a variety of modeling techniques can predict corrosion with reasonable accuracy, regression trees are particularly effective in modeling the complex relationships between the eddy current measurements and the actual amount of corrosion.

General Terms: Algorithms, Performance.

Keywords:
Data Mining, Knowledge Discovery, Regression Trees, Corrosion.

1. INTRODUCTION

Many commercial and military aircraft have reached or exceeded their original design life and are subject to significant increases in maintenance and repair cost due to corrosion. Corrosion is now recognized to have a detrimental effect on the structural integrity of aging aircraft components, and the lack of predictive capability has prevented the operators of aging aircraft from successfully controlling corrosion. There is particular concern about potential catastrophic damage from corrosion on the structural integrity of the fuselage. Corrosion may lead to a decrease in strength as a result of a loss in skin thickness, early fatigue crack initiation caused by the formation of stress risers, and increased fatigue crack growth rates. [4]

While corrosion problems are endemic to all services and all commercial aircraft, the United States Air Force (USAF) has many old (20 to 35+ years) aircraft that are the backbone of the total operational force. The oldest are the more than 500 jet tanker aircraft, the KC-135s, which were first introduced into service more than 40 years ago. For the most part, replacements are a number of years away, and the program schedules continue to be constrained by, and subject to, the vagaries of annual funding cycles. The KC-135s, along with many aging aircraft, have no planned replacement and are expected to stay in service for another 25 years. [13]

With varying degrees, all USAF aircraft have encountered and will continue to show signs of fatigue, stress corrosion cracking, corrosion, and wear. Historically, corrosion has caused an escalation of maintenance costs and, in many cases, has severely impacted operational readiness due to the increased time required in depot level repair.

Corrosion is life threatening and costly. More efficient, inexpensive corrosion prediction, detection, and classification tools are desperately needed to protect civilian industry and the military from catastrophic accidents and overwhelming expenses.

"Corrosion control can be one of the aircraft industry's most effective weapons in the battle against airplane structural failures. Left undetected and/or untreated, corrosion can totally undermine the integrity of an aircraft and make it unsafe to fly. It is a problem that is not always acknowledged or easily solved, and constant vigilance is necessary." [14]

Corrosion costs are extremely high. The United States spends almost $300 billion a year [14], the North American aircraft industry spends $13 billion a year [10] and the United States Air Force spends approximately $1 – 3 billion a year [13] on operations pertaining to corrosion costs. These monies for corrosion repairs and prevention programs take away from needed equipment upgrades and other operational programs. Due to budgetary constraints in both commercial and military sectors, there is a need for an efficient way to defend against the corrosion threat.

Current methods of corrosion detection, mainly non-destructive tests, rely on trained operators to find corrosion and other flaws. Hence, maintenance decisions based on these tests are highly dependent on the analytical prowess of the operator. "Human Factor studies applied to NDT [Non-destructive Testing] in aircraft maintenance facilities have shown that there is a large variety of factors that influence inspector performance. Many of these factors apply to other NDT methods as well, but with increasing degrees of automation, the effect of these factors can be reduced and the reliability of the inspections is expected to improve." [6] Table 1 provides a list of the factors that influence the ability of an operator to detect corrosion.

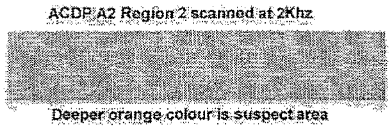
Table 1: Probability of Flaw Detection Considerations [2]

The probability of flaw detection is based on many considerations, e.g.
1. Operator training, alertness and confidence
2. Correct application of proper technique
3. Environment of the test – laboratory or field
4. Material homogeneity and isotropy
5. Flaw characteristics
6. Shape of part
7. Calibration and capability of the system
8. Other factors

Table 1 shows that operators skills at the corrosion identification task can be degraded by a range of factors. [2] For example, inappropriate training, lack of sleep, or simply lack of focus can result in miscalculation of corrosion damage. Boredom is a key factor when conducting non-destructive evaluations since the likelihood of finding a flaw is typically small. Thus, the number of times actual corrosion is detected is strongly outweighed by the times it is not. [2]

The current approach to corrosion detection creates a false-color image of the measurements from the non-destructive tests (Figure 1). It is then up to the operator to examine the image and identify defects in the material. The difficulty in interpreting this visual display of the measurements is whether the clarity, color, and detail of the visualization are sufficient to make a determination of flawed materials. There is also the question of how the data were filtered or manipulated to create this visualization and whether that introduced additional errors.

Figure 1: Visual representation of eddy current response [7]


ACDP A2 Region 2 scanned at 2Khz
Deeper orange colour is suspect area

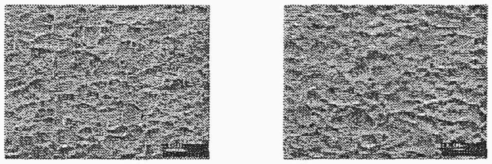
Improper representation of data by choosing the wrong resolution of the image or an inadequate color palette can lead to a wrong conclusion. Because the eye has a non-linear response to color, the perception of color varies from person to person, thereby making the selection of an appropriate color scheme extremely difficult. [8] In the case of corrosion detection, a miscalculation of whether a surface is flawed may have a catastrophic result.

2. ARTIFICIAL AND NATURAL CORROSION

In order to conduct our study to discover a relationship between NDT data and corrosion, we need precise measurement of the extent of the corrosion. Thus we have built models based on artificial corrosion, where the value of the material loss is known. With the improvement of artificial corrosion production, non-destructive tests have shown that the raw results from scans on artificial corrosion are similar to tests conducted on natural corrosion. Artificial corrosion plates are often used as controlled calibration specimens, in order to ensure proper operation of the scanning equipment.

The Institute for Aerospace Research, Canada developed an accelerated process to simulate the corrosion products and damage associated with crevice corrosion, which typically occurs in lap joints. The corrosion specimens formed during the accelerated process were very similar to those found in naturally corroded lap joints. In addition, the artificial corrosion damage had similar characteristics to that developed during the natural process. [10]

Figure 2: Artificial and Natural Corrosion [10]



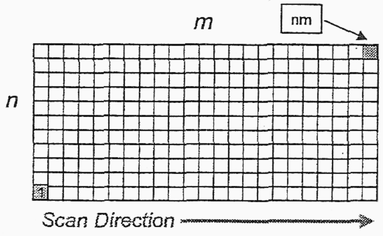
3. DATA ACQUISITION

The data used for this paper were acquired from the Institute of Aerospace Research (IAR), National Research Council, Canada. The datasets and related information are from a funded study by the United States Air Force. The study, titled "Nondestructive Inspections of Calibration Specimens and KC 135 Aircraft Specimens: NRC-LTR-ST-2267," was used to test the performance of several methods of non-destructive tests with a focus on eddy and pulsed eddy current testing.

The specimens that were tested included artificial corrosion calibration specimens and retired KC-135 aircraft parts. For the eddy current tests, the specimens were scanned using a multi-frequency probe. Each specimen was scanned using four frequencies: 5.5 kHz, 8 kHz, 17 kHz, and 30 kHz for 0.04-inch panels and 2 kHz, 4 kHz, 7 kHz, and 12 kHz for 0.063-inch thick panels. These panels were scanned using both eddy current and pulsed eddy current non-destructive testing techniques. This paper will focus on the eddy current scans.

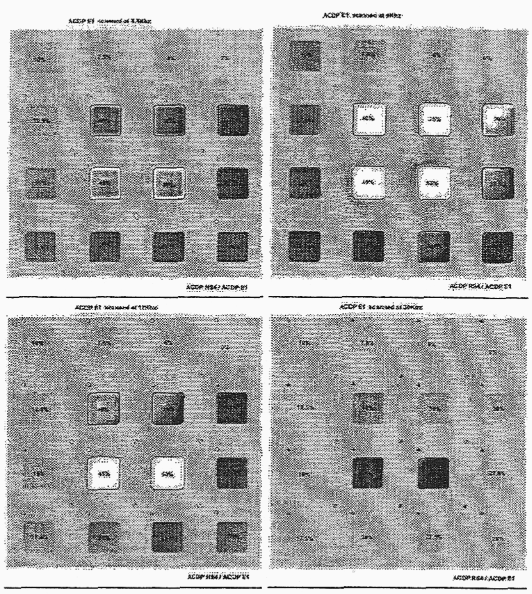
Each eddy current test produced four different data files, one for each scan frequency. The scans were conducted left to right from the bottom left corner of the specimen to the top right. Each scan point produced one data point as shown in Figure 3. The data points were voltage measurements that included negative values.

Figure 3: Scan pattern



Included with the eddy current datasets were bitmap images that visually represent the areas of material loss. These images were created from the response of the eddy current scans. When combined with calibration specimens these pictures are very useful, mainly because the corrosion areas are known and can be colorized for differentiation; it gets more difficult when the corrosion areas are not known. These bitmaps were a key element in the data mapping phase of the training data in this study. The eddy current scan of the calibration specimen E1 is shown in Figure 4. From top left to bottom right are the 5.5 kHz, 8 kHz, 17 kHz, and 30 kHz scan results.

Figure 4: Calibration Specimen E1 visual results [7]



4. DATA MAPPING AND CONSISTENCY

The term data mapping is used to describe the action of combining four scan files (predictor or input variables) and determining the associated material loss, which was the response variable's value. We also performed consistency checks by comparing our resulting data set with the one used in the IAR study.

The first step was to decode the given files from the original data format and create new files containing a single column of *nm* observations. These files were then combined into a single file with four columns and *nm* rows.

The next step was to add the response variable's values – the amount of material loss at a given location. These values were found by mapping the contents of the image to the appropriate frequency observations.

A program (Picview) was created to read the bitmap images and apply numerical values 0 or 255 to each observation based on a user chosen threshold. The numerical values were assigned to the red, green, or blue spectrum. Different combinations of the numerical values created a different color response. Files of different thresholds were used as an added measure to ensure proper response mapping. The starred areas in Figure 5 show the material loss areas each image was responsible for generating.

Figure 5: Picview Images

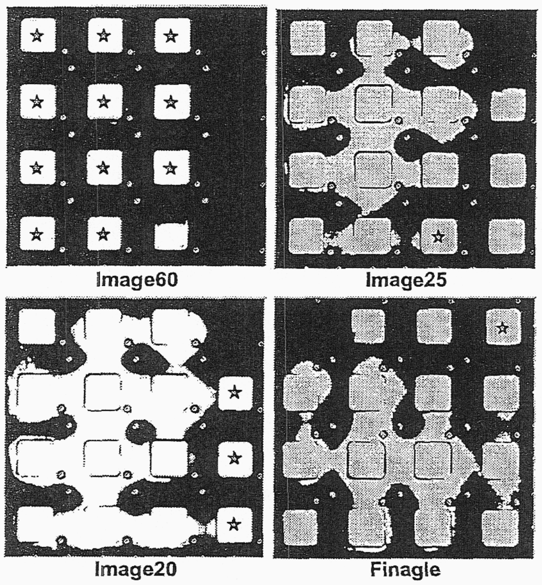


Image60

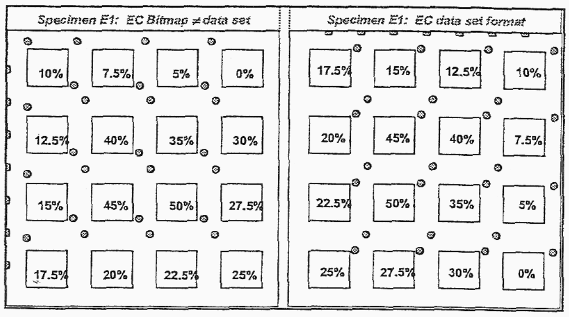
Image25

Image20

Finagle

The original datasets were supposed to have the data points in a corresponding order to the bitmap images provided (Figure 4). The data mapping process would have been an easy task if this supposition were true. However, mapping the dataset quickly became a puzzle. A visual comparison of what the dataset should have looked like and the actual mapping scheme is shown as Figure 6.

Figure 6: Bitmap image vs. actual data layout



After the above conversions of the total dataset, the training set was constructed using the actual data layout and Picview image data. Only the loss areas, represented by the labeled squares in Figure 6, were used in the final dataset; the rest of the data were deleted. The original dataset had 606,825 data points; the new dataset has 160,608 observations. Paring down this dataset deletes many noisy data elements that have no consequence on the results.

Once we had generated the dataset for the study, we validated this set. Our validation step used the graph shown in the original study by IAR and reproduced in Figure 7. Since the raw data used in the original graph were not available; we could only validate our data set by comparing our results with those shown in this graph (Figure 7). The graphs were compared by looking at the values of frequency response for each value of material loss. All voltages used in the graphs were average values.

Figure 7: Original graph from LTR-ST-2267 [7]

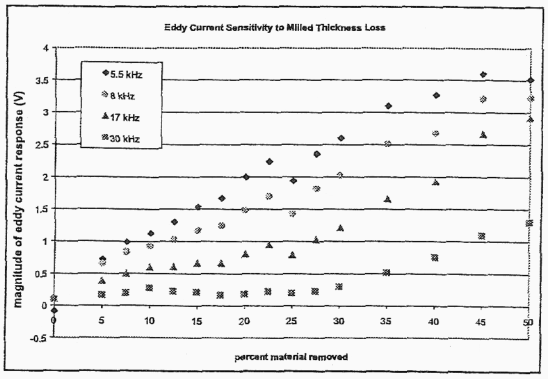
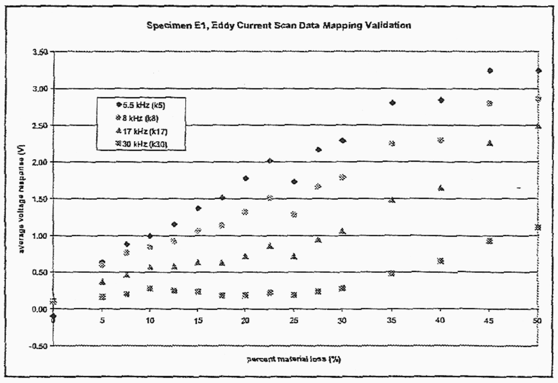


Figure 8: Graph from data mapping



The curve of each frequency scan was comparable; however, Figure 8 had smaller average values in general. The difference in magnitude between the associated graphs was considered inconsequential, since the difference was uniform throughout the responses of the various frequencies. Hence, our results would require at most only a translation of the frequency response to obtain their exact results.

5. MODEL EVALUATION

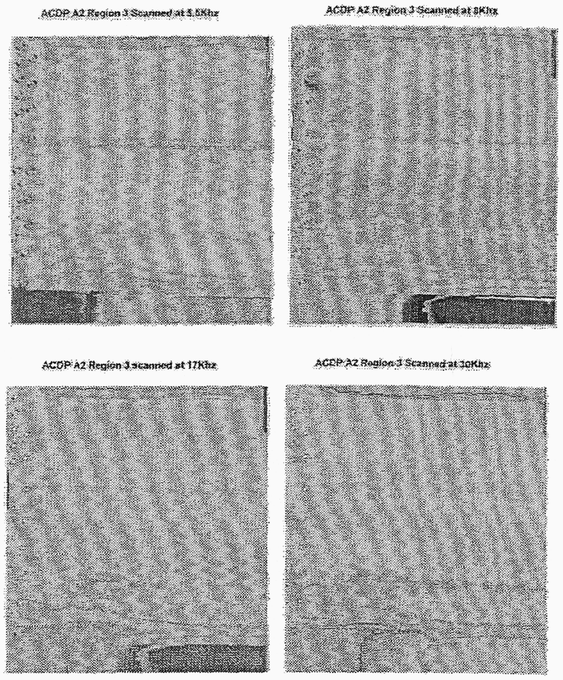
We sought to evaluate models from the data mining process using a test set. In particular we consider using a dataset created from scans on natural corrosion. Unfortunately, there was no way to validate the actual material loss. Additionally, the scans from the IAR study proved inconclusive in exposing material loss (Figure 9). Note the bottom fourth of these scans is unusable anyway. Once again, from top left to bottom right are the 5.5 kHz, 8 kHz, 17 kHz, and 30 kHz scan results.

Without a natural corrosion test dataset, we chose to evaluate models using a random split of 75% (120,456 observations) of the specimen E1 data for training and 25% (40,152 observations) for testing. The model-building portion of the dataset needed to be sufficiently large so that a reliable model could be developed. In order to ensure robustness of the training model, the test dataset applied was also kept fairly large.

6. APPROACHES USED FOR DATA MINING

In order to search for relationships between NDT data and corrosion we had a choice from a variety of modeling techniques. Essentially the problem has ratio scaled predictor variables and an interval scaled response variable. Several methods have direct application and others can be applied, even though their underlying assumptions do not strictly hold. For example, multiple linear regression, regression trees, group method of data handling (GMDH) "polynomial networks," and ordinal logistic regression are the methods we used for data mining in this particular case.

Figure 9: Specimen A2, Region 3 [7]


ACDP A2 Region 3 Scanned at 5.5Khz
ACDP A2 Region 3 Scanned at 8Khz
ACDP A2 Region 3 scanned at 17Khz
ACDP A2 Region 3 Scanned at 30Khz

In general, more complex models tend to provide excellent accuracy with training data but do poorly with test data. Usually this means that a model has over-fit the data and the model will fail miserably with the application of fresh data in a real setting. One reason for this phenomenon is that complex models tend to over-fit the noise in the training data set, which then does not model the true process in the test data (or the real data). Hence, data miners tend to apply Ockham's razor, and choose simpler models if the accuracy of the result in the test data does not improve very much. Surprisingly for the data in this problem we found that complexity was good in both training and test sets.

In both multiple regression and ordinal logistic regression, the models were 4th order polynomials with interaction terms. Statistically, the more parsimonious models did not perform as well. However, the data set did show heteroscedasticity, so a heteroscedastic-consistent covariance matrix (HCCM) was used to protect against heteroscedasticity for the multiple regression model.

The GMDH model (more complex still, with the equivalent of an eighth degree polynomial) performed better than both the multiple regression and ordinal logistic regression models (see Table 2 test set results). However, as shown by the values of root mean square error, the improvement was slight.

Both least squares (LS) and least absolute deviation (LAD) regression tree splitting methods were tested. The least squares tree performed slightly better than the other methodologies with 1,857 nodes, but a better choice is the least absolute deviation model with only 819 nodes. The LAD regression tree model significantly outperformed the other models.

Table 2: Comparison of methods using a test dataset

Overall Model Comparision by Test Set		
Model	*RT MSE*	*VAR*
Multiple Regression Model 8	5.388	29.030
Logistic Regression Model 8	5.610	31.468
GMDH Polynomial Network	4.872	23.739
LAD Regression Tree	***0.566***	0.320

7. INTERPRETATION OF RESULTS

The complexity of the models discovered with the seeming violation of Ockham's razor was an interesting occurrence in this study. In order to explore this phenomenon, a three-dimensional graph was developed using the three most important predictor variables, 5.5 kHz, 8 kHz, and 17 kHz. This graph shows the complex nature of the data where there are "patches" of response values strewn throughout the three-dimensional plane. The different colors on the graph show the various material loss values.

Figure 10 shows why the more complex models and tree-based models statistically performed better than the other models. These models tend to "stitch" the data together in order to estimate the material loss. Notice the several "patches" of color that could be segregated to provide better modeling accuracy.

Figure 11 shows how stitching occurs in the two dimensional case. Notice that each response follows a different dispersal pattern. The simpler parametric model would not be able to adequately estimate these responses. Therefore, the more complex a parametric model, the better it will perform, even with the test dataset. A non-parametric algorithm performs well, because such models do not fit a curve to the data; they use cuts to split out the response.

Figure 10: Two perspectives of the eddy current data

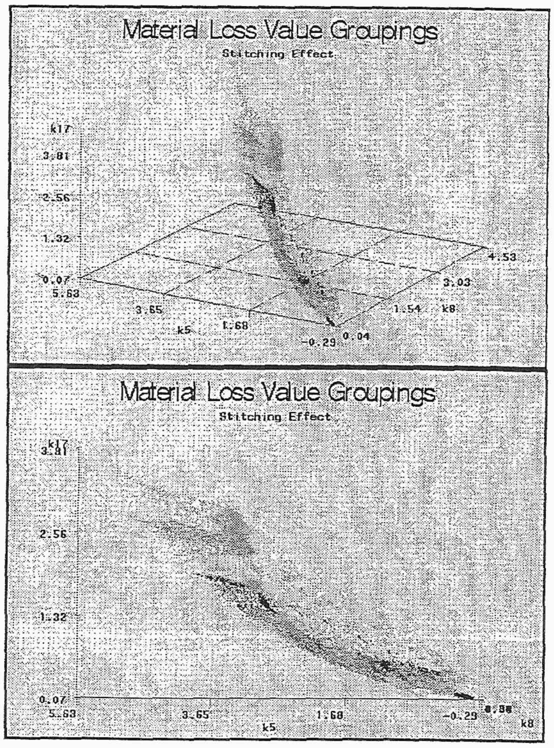
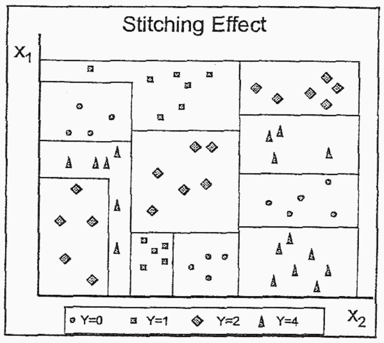


Figure 11: Example of Stitching Effect in 2D



8. CONCLUSIONS

This paper has described the use of data mining to discover relationships between NDT and corrosion. We have shown the potential to augment current methods of visually displaying NDT data and asking maintenance operators to find significant corrosion. In particular, we have found models that predict corrosion with average errors of about 0.6 % of total material loss. The best models derive from regression trees that split the NDT data into cells that correspond to the different values of material loss.

Every model tested showed a surprising level of complexity that can be attributed to the nature of the corrosion and the NDT testing.

Corrosion occurs at a very fine scale, and moves through the metal in ways that are determined by a variety of factors. For example, two important factors are environmental conditions experienced by the aircraft and characteristics of the manufacturing processes that produced both the metal and the aircraft. Eddy currents are generated by moving an induction coil over local areas that are either free of corrosion or corroded, which causes non-linear jumps in their values over very small regions of space. The combinations of eddy currents at different frequencies can help to isolate corrosion in small areas but to do this requires complex models that can handle the inherent non-linearities. Regression trees, which are non-linear, seem well suited to this local isolation problem. When we look at figures 10 and 11, we see that the separate small regions that contain the different values of corrosion. In empirical modeling terms, this is a multi-modal problem, where the modes for each material loss value are separated in feature space into many small regions. Hence, data mining has effectively uncovered the complexity of the corrosion- NDT relationship.

Additional work will include analysis of data sets with other NDT measurements to augment the eddy data. We expect to obtain these data sets over the next year. The work reported here will provide a basis for mining these sets and looking for broader relationships between NDT and corrosion. The work so far provides a good basis building systems that can support maintenance operations and significantly reduce the chances of missing corrosion that may lead to catastrophic failures.

REFERENCES

[1] Agresti, Alan., An Introduction to Categorical Data Analysis., John Wiley & Sons, Inc., New York, 1996.

[2] Bray, Don E. & Stanley, Roderic K., Nondestructive Evaluation: A Tool in Design, Manufacturing, and Service. CRC Press, Boca Raton, 1997.

[3] Breiman, L., Friedman, J., Olshen, R. A. & Stone C. J., Classification and Regression Trees. Wadsworth and Brooks/Cole, Monterey, CA, 1984.

[4] CC Technologies Laboratories Inc., Mathematical Model to Predict Fatigue Crack Initiation in Corroded Lap Joints, Draft. Dublin Ohio. August 1998.

[5] Farlow, Stanley J., Self-Organizing Methods in Modeling: GMDH Type Algorithms., Marcel Dekker, Inc., New York, 1984.

[6] Forsyth, D. S., Automation of Enhanced Visual NDT Techniques., www.ndt.net/article/pacndt98/18/18.htm

[7] Forsyth, D.S., Nondestructive Inspections of Calibration Specimens and KC 135 Aircraft Specimens. Institute for Aerospace Research, Canada, 2000.

[8] Gros, X.E. , NDT Data Fusion ., John Wiley and Sons., New York, NY 2000.

[9] Hosmer, David W. & Lemeshow, Stanley., Applied Logisitc Regression., John Wiley& Sons, Inc., New York, 2000.

[10] IAR (Institute for Aerospace Research)., http://www.nrc.ca/iar/corrosion_e.html , Canada, 2001.

[11] Long, J.S. and Ervin, L., "Using Heteroscedastic Consistent Standard Errors in the Linear Regression Model"., American Statistician, 54, pp 217-224., 2000.

[12] Madala, Hema R. & Ivakhnenko, Alexy G., Inductive Learning Algorithms for Complex Systems Modeling., CRC Press Inc., Boca Raton, Florida., 1993.

[13] The National Academy Press, Aging of U.S. Air Force Aircraft (1977). www.nap.edu/books/0309059356/html/index.html.

[14] NACE International. "NACE Issue Paper: Aircraft Corrosion". http://www.nace.org/naceframes/Government/planefnl.htm , 2000.

Damage Prediction and Estimation in Structural Mechanics Based on Data Mining [*]

S. S. Sandhu, R. Kanapady and K. K. Tamma
sandhu@me.umn.edu, ramdev@me.umn.edu and ktamma@tc.umn.edu
Dept. of Mechanical Engineering, University of Minnesota, 111 Church St. S.E, Minneapolis MN 55455

C. Kamath
kamath2@llnl.gov
Center for Applied Scientific Computing, Lawrence Livermore National Laboratory, Livermore, CA 94551

V. Kumar
kumar@cs.umn.edu
Dept. of Computer Science and Engineering, University of Minnesota, 200 Union Street S.E, Minneapolis MN 55455

ABSTRACT

Damage in a material includes localized softening or cracks in a structural component due to high operational loads, or the presence of flaws in a structure due to various manufacturing processes. Methods that identify the presence, the location and the severity of damage in the structure are useful for non-destructive evaluation procedures that are typically employed in agile manufacturing and rapid prototyping systems. The current state-of-the art techniques for these inverse problems are computationally intensive or ill conditioned when insufficient data exists. Early work by a number of researchers has shown that data mining techniques can provide a potential solution to this problem. In this paper, we investigate the use of data mining techniques for predicting failure in a variety of 2D and 3D structures using artificial neural networks (ANNs) and decision trees. This work shows that if the correct features are chosen to build the model, and the model is trained on an adequate amount of data, the model can then correctly classify the failure event as well as predict location and severity of the damage in these structure.

1. INTRODUCTION

Damage in a material includes localized softening or cracks in a structural component due to high operational loads, or the presence of flaws in a structure due to various manufacturing processes. Methods that identify the presence, location and the severity of damage in the structure are useful for non-destructive evaluation procedures that are typically employed in agile manufacturing and rapid prototyping systems. In addition, these techniques will be critical to reliable prediction of damage to bridges, skyscrapers and structures deployed in space.

Damage detection involves three stages of characterization. First, whether the damage has taken place in the structure (recognition); second, where the damage has taken place in the structure (location); and finally, the severity of the damage in the structure (quantification). Structural damage results in changes in structural responses such as static displacements and dynamic properties such as natural frequency, and the mode shapes of the structure. Although rigorous damage models exist, in this work we focus on the structural damage that is assumed to be associated with structural stiffness as a reduction in Young's modulus (E) [1].

A practical damage assessment methodology must be capable of predicting structural stiffness as a function of changes in structural response and dynamic properties [2]. Standard analytical techniques employ mathematical models to approximate the relationships between specific damage conditions and changes in the structural response or dynamic properties. Such relationships can be computed by solving a class of so called inverse problems [3, 4]. The current state-of-the art techniques for these inverse problems are computationally intensive or ill conditioned when insufficient data exists.

Early work by a number of researchers [1, 2, 5, 6, 7] has shown that data mining techniques can provide a potential solution to this problem. These efforts have focussed on employing ANNs to predict damage using static displacements and dynamic properties. However, these studies only consid-

[*] The authors are very pleased to acknowledge support in part by the Department of Energy DOE/LLNL W-70450-ENG-48 and by the Army High Performance Computing Research Center (AHPCRC) under the auspices of the Department of the Army, Army Research Laboratory (ARL) cooperative agreement number DAAH04-95-2-0003/contract number DAAH04-95-C-0008. The content does not necessarily reflect the position or the policy of the government, and no official endorsement should be inferred. Access to computing facilities was provided by AHPCRC and Minnesota Supercomputer Institute (MSI)

ered small scale plane structures in two dimension. Furthermore technical details related to selection of features, training and testing data sets etc, were not investigated in detail. In this paper, we investigate the use of data mining techniques for predicting failure in a variety of two dimensional (2D) and three dimensional (3D) structures using artificial neural networks (ANNs) and decision trees. ANNs approach is attractive in that it can learn complex, highly nonlinear relationships, and can be used to solve inverse problem. On the other hand decision tree models are easy to understand and have the potential to discover useful rules. This work shows that if the correct features are chosen to build the model, and the model is trained on an adequate amount of data, these model can correctly predict the location and severity of the damage in these structure.

This paper is organized as follows. In Section 2, the problem statement and generation of the data to build data mining models is discussed. In Section 3, the data mining models using static displacements are built and evaluated. In section 4, dynamic properties of structures are used to build and evaluate data mining models. Section 5, presents conclusion and suggestion for future work on this topic are discussed.

2. PRELIMINARIES

2.1 Problem statement and description of data mining models used

The goal is to construct data mining models that can predict the Young's modulus (E) of the elements in the structure as a function of static displacements and dynamic properties.

We use ANNs developed by Rumelhart and McCelland [8] and, decision tree algorithms based on the work of Ross Quinlan (1993) to build predictive models for Young's modulus. Finding a suitable architecture of ANN for the problem is non trivial. All the ANN models built in this study have two hidden layers each employing roughly 20 nodes each. In this study, decision tree models are build using the algorithm provided in Clementine software[1], and ANN models are build using Matlab's ANN toolbox[2].

2.2 Generating the data

To build the right data mining model it is important that useful features are considered. The selected features should possess the property of correctly identifying damage states and should capture the physics of the problem at hand. The data is generated by using a finite element analysis code. The data layout is shown in Table 1 where $\mathbf{f} = \{f_1, \dots, f_n\}$ is the feature set and $\mathbf{E} = \{E_1, \dots, E_n\}$ represents the target variables where each record in the table pertains to a failure state. Each failure state is simulated by failing either one (single element failure) or more elements (multiple element failure) in the structure, in steps (e.g. failing each element by reducing E from the base value of E to E' in steps of ϵE where ϵ is a small fraction). Such simulations give the structural response such as the static displacements (at the nodes) and dynamic properties such as the natural frequencies of the structure. This data can then be used directly to train and test the data mining algorithms. New features can be derived from these raw features. In some cases, they lead to a better predictive model.

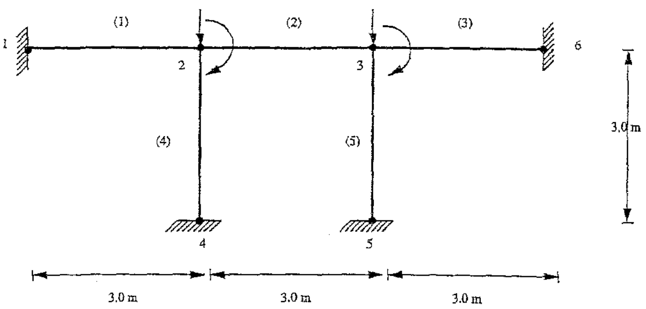


Figure 1: Plane frame structure discretized using beam elements.

	Features				Target variable			
S.No	f_1	f_2	...	f_n	E_1	E_2	...	E_n
1	72.833	151.67	...	213.45	0.5E	E	...	E
2	73.45	152.56	...	213.65	0.6E	E	...	E
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
500	74.01	153.01	...	214.21	E	E	...	E

Table 1: A typical input to the data mining model.

3. BUILDING DATA MINING MODELS USING STATIC DISPLACEMENTS OF STRUCTURE AS FEATURES

In this section, data mining models are developed by considering the static displacements at the nodes of the structure as features. Various examples with increasing complexity are considered to study the performance of data mining techniques.

2-D Structure – plane frame: The first structure used to build the data mining model is shown in the Fig. 1. It is a plane frame studied in [5] with the loads as shown. The nodes 1, 4, 5 and 6 are fixed and the nodes 2 and 3 are subjected to loads. During the generation of the data the loads are kept constant. Absolute static displacements namely $|u_2|$, $|v_2|$, $|\theta_{y2}|$, $|u_3|$, $|v_3|$, $|\theta_{y3}|$ (instead of raw data of displacements at nodes) of the nodes 2 and 3 were selected as the features. It was seen that selecting the absolute value of the nodal displacement leads to a better model, because changes in stiffness influence the magnitude of the displacements and not their sign.

The testing and training data set of 500 damaged states is generated by failing each element at a time. The value of E is varied from 0.01E to 0.99E in steps of 0.01E. The ANN is built by training it on a random sample of 60% of this data. The results for ANN are shown in Table 2. For this simple problem the models built by ANN are accurate, as the features considered are enough to accurately predict the

[1] ©1999 SPSS Inc., Version 5.0.1.

[2] ©1984-2000 The MathWorks, Inc. Version 6.0.0.88 Release 12.

target variable. A plot of the predicted value of E versus actual value of E for some typical element is shown in Fig. 2. Figure 2 shows an almost linear correlation between the predicted and actual E. From this, it is evident that the neural network can effectively predict the value of Young's modulus, and consequently the damage for this simple structure.

	E_1	E_2	E_3	E_4	E_5
e_r (%E)	0.032	0.04	0.082	0.076	0.026
σ (%E)	0.20	0.22	0.39	0.50	0.18
r	0.9999	0.9999	0.9998	0.9998	0.9999

Table 2: Result of testing ANN with absolute value of displacement as features for plane frame shown in Fig. 1,where e_r = mean relative error, σ = standard deviation, r = linear correlation.

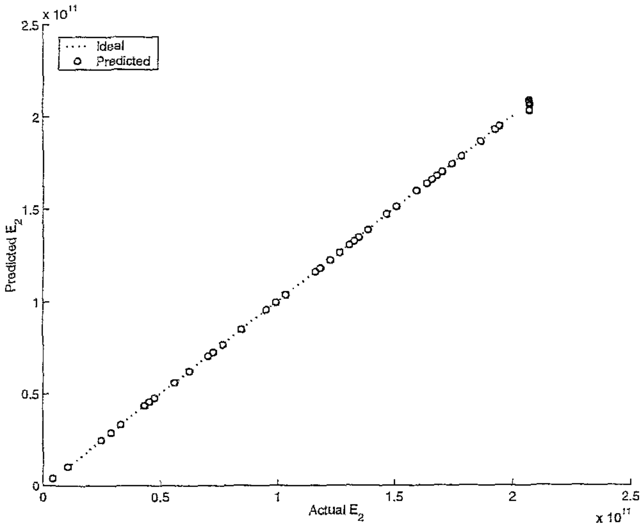


Figure 2: Comparison between ideal and actual E for typical element 2 for plane frame shown in Fig. 1.

To employ the decision tree algorithm the target variable E needs to be discretized. Hence in this case the value E was restricted to : i) 0 - severely damaged, ii) 1 - moderately damaged, and iii) 2 - undamaged. The data for training and testing the decision tree model is generated in exactly the same manner as that for the ANN. The decision tree is trained on 60% of the data generated and tested on the entire data. The result obtained on testing the decision tree is shown in the form of coincidence matrix (which shows the number of damage states that have been classified correctly and incorrectly) in Table 3. Since the coincidence matrix is predominantly diagonal, the model build by using decision tree is highly accurate.

Predicted E's							
		E_2			E_5		
		0	1	2	0	1	2
Actual E's	0	72	0	0	72	0	0
	1	0	19	0	0	16	3
	2	0	0	409	0	0	409

Table 3: Coincidence matrix with absolute value of displacement as features for plane frame shown in Fig. 1.

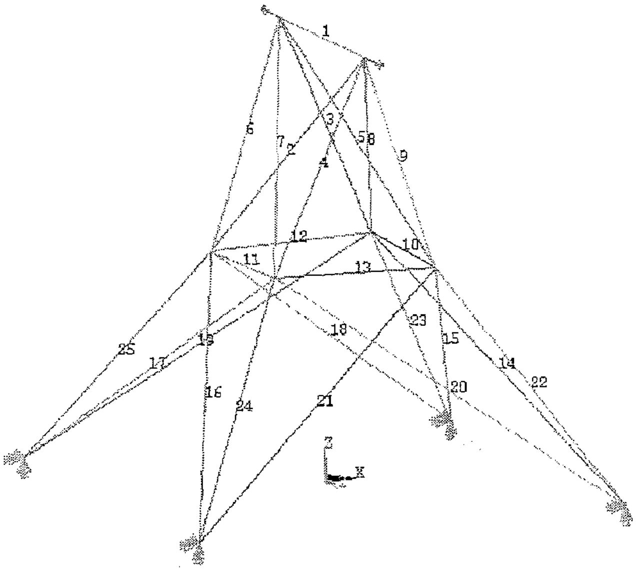


Figure 3: Three dimensional electric transmission tower discretized using beam elements.

3-D Structure – electric transmission tower: The second structure we consider is a 3-D electric transmission tower. This structure shown in Fig. 3, consists of beam elements oriented in 3-D space. The transmission tower consists of 10 nodes out of which the representative transmission cable loading is applied at nodes 3 and 4. The nodes 7, 8, 9 and 10 are fixed to the ground. In this case, unlike the case of the plane frame, each node and element has displacements in all three direction (u, v, w), together with bending about two axes (θ_y, θ_z) and torsion about the axis of the beam (ϕ_x). These are commonly referred to as the degrees of freedom at any point in the structure. Due to the complexity of the structure, the problem is non trivial. To develop an adequate data mining model, a significantly large number of damage states are required. The study is conducted with two different sets of features. In one set of features, the absolute value of the displacements at the nodes is used. Hence there exist 36 features for each damage state. Another set of features are defined as follows. For any element e defined by nodes i and j, the element displacement measures are defined as

$$d_e = \sqrt{(u_i - u_j)^2 + (v_i - v_j)^2 + (w_i - w_j)^2} \quad (1)$$

$$\theta_{e1} = |\theta_{yi} - \theta_{yj}| \quad (2)$$

$$\theta_{e2} = |\theta_{zi} - \theta_{zj}| \quad (3)$$

$$\phi_e = |\phi_{xi} - \phi_{xj}| \quad (4)$$

where d_e is a measure of the element translation, θ_{e1} and θ_{e2} are measure of element bending and ϕ_e is a measure of element torsional displacement. Hence, there are a total of 100 features in this case.

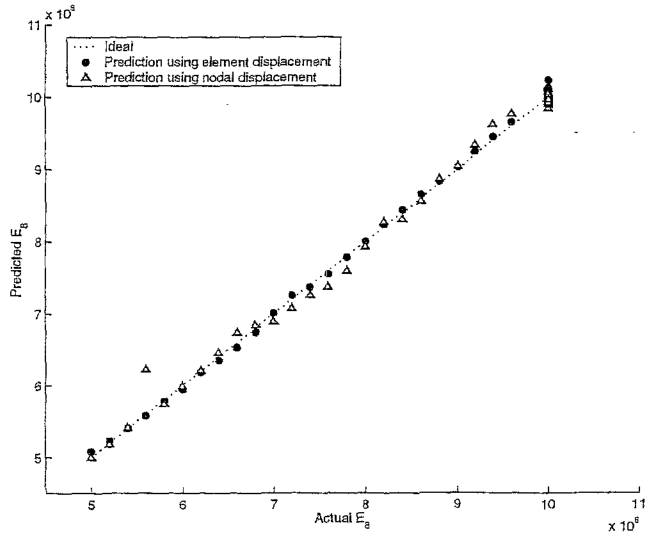


Figure 4: Comparison between ideal and actual E for typical element 8 for electric transmission tower shown in Fig. 3.

The testing and training data are generated with each element of the structure being damaged by reducing the value of E to 0.5E in steps of 0.02E leading to 600 damage states. Both ANNs and decision trees are trained using 70% of the total generated damage states. Figure 4 shows the comparison between the test results of the model using displacement of the nodes and the model using element displacement measure as features. It is evident from the figure that the model using the element displacement measure features (Eqs. 1 – 4) is more accurate.

Predicted E's							
		E_1			E_4		
		0	1	2	0	1	2
Actual E's	0	13	0	0	0	0	13
	1	0	9	1	0	0	10
	2	7	0	570	0	0	577

Table 4: Some typical coincidence matrices with absolute displacement of nodes as features for transmission tower shown in Fig. 3.

Predicted E's							
		E_1			E_4		
		0	1	2	0	1	2
Actual E's	0	13	0	0	13	0	0
	1	0	10	0	1	9	0
	2	0	0	577	0	0	577

Table 5: Some typical coincidence matrices with elemental displacement measures as features for transmission tower shown in Fig. 3.

Tables 4 and 5 show the results for decision trees with absolute nodal displacement and element displacement measure features respectively. Again, it can be clearly seen that the element displacement measure prove to be better features for decision tree models. Unlike neural networks, the models developed by decision tree can be readily understood and interesting rules can be found. For example, a rule generated in this case is given by

```
if $\theta_{12} \leq 0.185$ then
        if $\theta_{10} \leq 0.09$ then
                if $\theta_7 \leq 0.545$ then $E_3 = 2$
                else $E_3 = 1$
        else $E_3 = 2$
else if $\phi_3 \leq 0.083$ $E_3 = 2$
else $E_3 = 0$
```

This rule says that the failure of element 3 depends on the displacement of elements 7, 10 and 12 which are connected to element 3 (Ref. Fig. 3). Such interesting rules not commonly known in the traditional analysis community can be discovered which can be potentially useful to a structural designer.

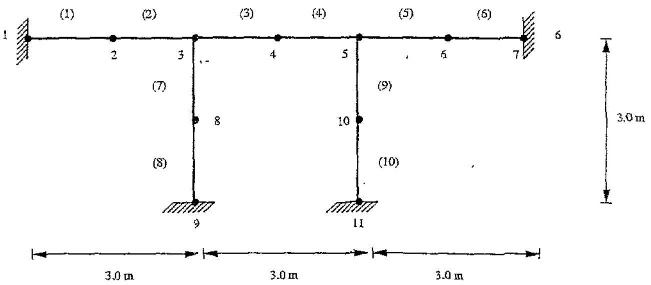


Figure 5: Plane frame discretized using beam elements.

Static displacements with varying loads: In the cases considered previously the data mining models were built using constant loading. But many structures are subject to variable loading (when the loads the structure is subjected to are continuously changing) and so an effective model should be able to correctly predict damage in this case. Although the static displacement features are not load independent, in this section their performance is studied when they are used to build a model for predicting failure under variable loading conditions. The plane frame structure in Fig. 5 is used to build the model. The feature set consists of features which correspond to the location and magnitude of the loads in additional to the static displacements of nodes 2 and 3. Three different loading conditions are considered. First, node 2 and 6 are loaded. Next, node 3 and 5 are loaded. Finally, node 8 and 10 are loaded.

The training data is generated by failing each element in the structure by reducing its Young's modulus from 1.0E to 0.5E in steps of 0.1E. The loads in each of the three loading conditions considered are varied from 500N to 2500N in steps of 500N. The testing data is generated by failing each element in the structure by reducing its value of E from 0.95E to 0.45E in steps of 0.1E. The loads in each of the three loading conditions considered are varied from 250N to 2250N. This leads to 2250 failure states, each for testing and training the ANN. The test results are shown in Table 6. The plots of the predicted and actual E for a typical element 5, is shown in Fig. 6. From the results it can be seen that predicting capability of the model using static displacement reduces when the loads are varying, because two different loads corresponding to different failure states can produce the same response. Further investigations are necessary to rectify this situation.

	E_1	E_5	E_7	E_9
e_r (%E)	1.35	1.22	1.12	0.95
σ (%E)	2.06	2.24	1.90	1.76
r	0.98	0.98	0.98	0.99

Table 6: Result of testing ANN when a variation in loading is considered for plane frame shown in Fig. 5.

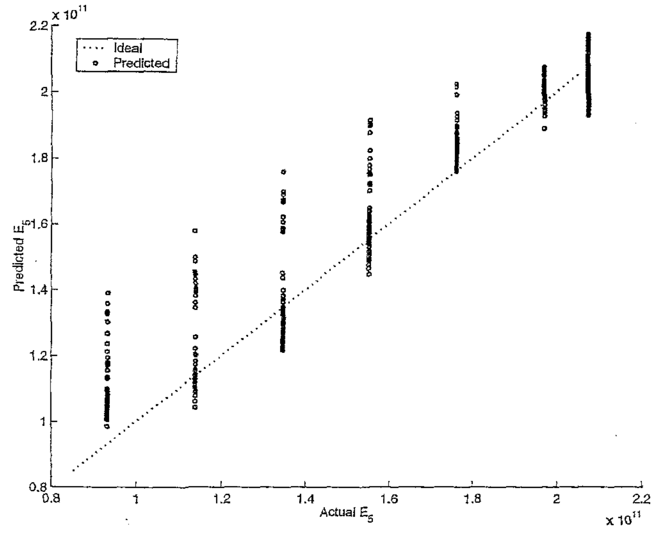


Figure 6: Comparison between ideal and actual E for typical element 5, when variation in the load is considered for plane frame shown in Fig. 5.

Failure of multiple elements: In the previous examples, the model is trained and tested to predict damage with only one element failure in the structure. This seems relevant because failure in the structure generally starts from one element and then spreads to other elements. Here the case when multiple elements of the structure have failed is discussed. For predicting damage in multiple elements of the structure, the plane frame structure used previously in Fig. 1 is employed. The set of features are again the displacement coordinates of nodes 2 and 3.

The data used for training is generated by reducing the Young's modulus of each of the elements simultaneously from E to 0.5E in steps of 0.1E. This results in $6^5 - 1 = 7775$ failure states and one undamaged state. After the data is generated both the ANN and the decision tree are used to build models for predicting the damage in the structure. For ANN 5% (395 failure states) of this generated data is randomly sampled for training. Testing data of 1000 failure states is generated by failing each element and choosing its E randomly. The results for the ANN are shown in Table 7.

	E_1	E_2	E_3	E_4	E_5
e_r (%E)	0.032	0.046	0.057	0.050	0.028
σ (%E)	0.040	0.063	0.074	0.064	0.037
r	0.9999	0.9999	0.9999	0.9999	0.9999

Table 7: Result of testing ANN when multiple elements are failed for plane frame shown in Fig. 1.

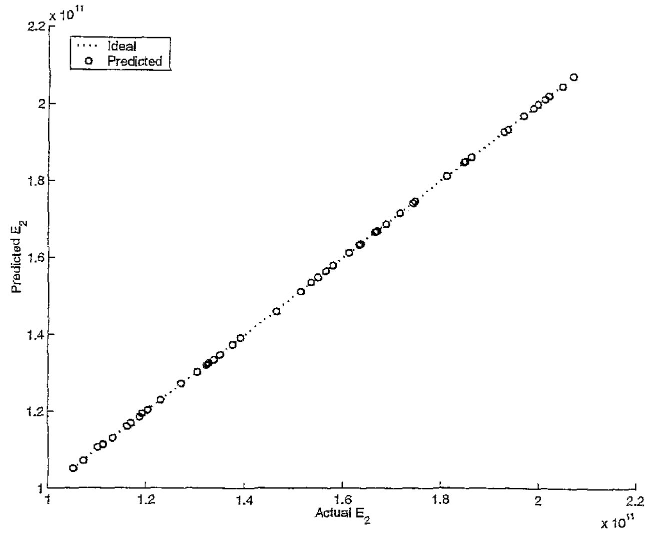


Figure 7: Comparison between ideal and actual E for typical element 2, when multiple elements of structure fail for plane frame shown in Fig. 1.

The plots of the predicted E versus actual E for a typical element 2, is shown in Fig. 7. It is evident that the correlation between them is almost linear. Hence, static displacements, prove to be effective features in building an ANN model to predict failure in multiple elements. In the case of decision tree the coincidence matrix shown in Table 8, is predominantly diagonal.

		Predicted E's					
		E_2			E_4		
		0	1	2	0	1	2
Actual E's	0	3771	116	0	3588	298	1
	1	97	2365	130	282	2016	294
	2	0	145	1151	2	416	878

Table 8: Some typical coincidence matrices for the case when multiple elements of the structure are failed for plane frame shown in Fig. 1.

4. BUILDING DATA MINING MODELS USING DYNAMIC PROPERTIES OF STRUCTURE AS FEATURES

Dynamic properties of the structure as features provides an alternative approach for predicting damage. Its advantages over using static displacements are:-

1. While different loads produce different static displacements, the dynamic properties of the structure are essentially load independent. For example, dynamic properties of the structure include natural frequencies and mode shapes.
2. In the case of static displacements, different components of the displacement at each node are used as features, which result in a large number of features for larger finite element discretizations. On the other hand if dynamic properties like natural frequency are used, then features in the form of only the lowest 'n' natural frequencies can be used resulting in a reduction of the number of features.

The natural frequency and the mode shape of the structure are obtained by solving the eigenvalue problem:

$$[-{\omega_i}^2\mathrm{M} + \mathrm{K}]\Phi_i = 0 \qquad (5)$$

where M and K are mass matrix and stiffness matrix, of the structure respectively, and ω_i is the natural frequency corresponding to the mode shape Φ_i. Damping in the structure has been neglected in this study. Structural damage results in changes in dynamic properties. The prediction of damage in the structure can be achieved if the model is taught to recognize the changes in the frequencies and the mode shapes with the failure of specific members in the structure. To train the ANN, the elements of the structure are failed one at a time by reducing their modulus of elasticity. For this failure state the natural frequencies and mode shapes are obtained by solving the eigenvalue problem in Eq. 5.

Natural frequency: 2-D Structure – three span bridge:

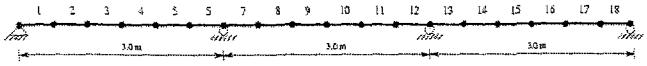


Figure 8: Three span bridge structure modeled using beam elements.

The structure used for this study is a three-span, equal length continuous beam, with constant properties that was studied in [5]. This structure is shown in Fig. 8. The beam is divided into 18 beam elements, with 6 equal length elements in each span. This structure is unsymmetric as regards to the boundary conditions. It is fixed at one end and simply supported at the other. The training data is generated by reducing the value of E from 1.0E to 0.5E in steps of 0.05E. The testing data is generated by reducing the value of E from 0.975E to 0.525E in steps of 0.05E. This results in the training and testing data of 181 and 180 records respectively.

The lowest 'n' natural frequencies of the structure $(\omega_1, \omega_2 \ldots \omega_n)$ are employed as features to predict damage. The study is conducted with a varying number of first 'n' natural frequencies. For the bridge structure studied here, the first 4 natural frequencies are adequate to build a fairly accurate predictive model. However, considering additional natural frequencies improved the accuracy of the model upto first nine natural frequencies. Further increase in the number of natural frequencies leads to a saturation and a slight deterioration in the model's performance. Table 9 and Fig. 9 shows the results for the model with lowest 9 natural frequencies.

	E_1	E_4	E_{10}	E_{14}	E_{17}
e_r (%E)	0.06	0.09	0.09	0.08	0.09
σ (%E)	0.106	0.13	0.17	0.12	0.15
r	0.999	0.999	0.999	0.999	0.999

Table 9: Result of testing ANN when natural frequencies are used as features for three span bridge shown in Fig. 8.

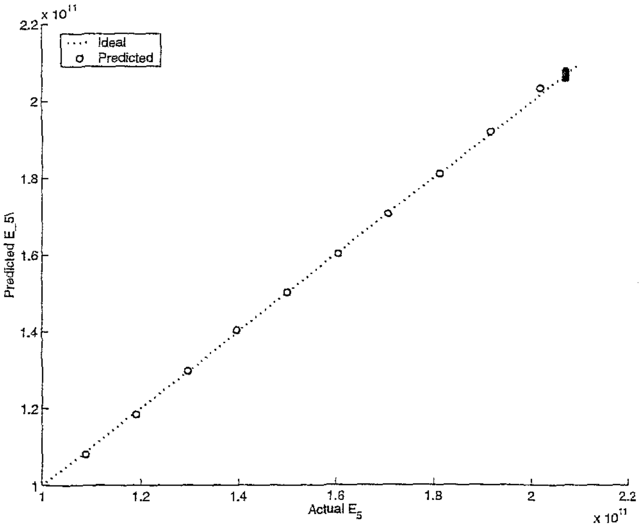
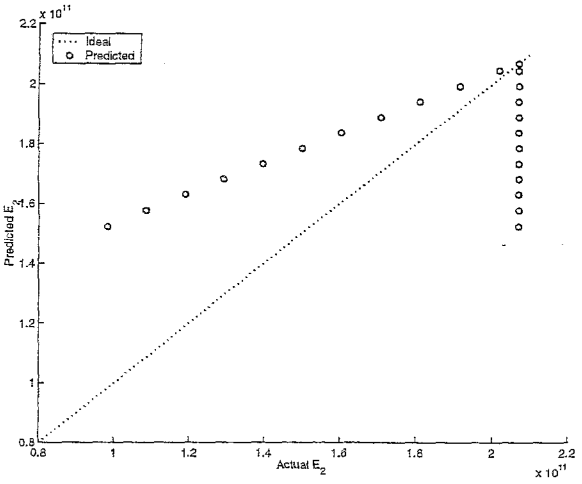


Figure 9: Comparison between predicted ideal and actual E when natural frequencies are used as feature for unsymmetric structure shown in Fig. 8.

Next, the structure in Fig. 8 is modified so that it is simply supported at both ends, to study the suitability of using natural frequency as features in case of structures exhibiting symmetry. The testing and training data is generated in the same manner as in the unsymmetric case. In Fig. 10, the results of testing the ANN for two symmetrically equivalent elements, element 2 and 17 is shown. In Fig. 10, the cases in which element 2 has not failed but has been predicted to have failed, corresponds to failure states when element 17 has failed and vice-versa. The same is the case for the other symmetrically equivalent elements. This is due to the fact that natural frequency is a global feature and, the change in the natural frequencies is the same, when either one of the symmetric elements is failed . The ANN has no reason



(a) Element 2

(b) Element 17

Figure 10: Comparison between predicted ideal and actual E when natural frequencies are used as feature for symmetric structure.

to favor the prediction of the failure of one element over the other. In order to keep the mean squared error, which is the performance criteria used to train the ANN to a minimum, the model predicts that both the elements have failed. The predicted value of the Young's modulus in this case is higher than the actual value of the failure, in order to keep the mean squared error a minimum. Thus, when a structure exhibits symmetry, using natural frequencies alone as the features is not sufficient and other dynamic features need to be considered or the structure may have to be modeled differently using symmetry considerations.

3-D Structure – electric transmission tower: Natural frequencies are used as features to predict damage in the structure shown in Fig. 3. The symmetry of the structure is disturbed by changing the cross-sectional area of the symmetric elements. The training data is generated by reducing the value of E from 1.0E to 0.5E in steps of 0.05E, generating a total of 251 records. The testing data set of 500 records, is generated by failing each element by an arbitrary amount. The first 12 natural frequencies were considered while building the model. The results of testing the ANN are shown in Table 10 and Fig. 11. The results show that the model is accurate in predicting the location and severity of the damage. Natural frequencies prove to be good features for predicting single element failure in an unsymmetric structure.

5. CONCLUDING REMARKS

This paper presented data mining models to predict the failure in the structure as a function of static displacements and dynamic properties. Damage was simulated by reduction in the values of Young's modulus of the elements in the structure. The prediction of the data mining technique greatly depends on the features chosen. A more meaningful attribute produces better results. Hence the data from the finite element analysis of the structure should be suitably preprocessed so that the raw data is converted into features that have a closer relationship with the target function. While using static displacements, new features such as absolute nodal displacements and elemental displacement measures were used to generate models for predicting failure. These features proved to be better than nodal displacements.

	E_1	E_4	E_{10}	E_{14}	E_{19}	E_{25}
e_r (%E)	0.13	0.17	0.11	0.15	0.15	0.20
σ (%E)	0.16	0.26	0.15	0.20	0.19	0.28
r	0.999	0.998	0.999	0.999	0.999	0.998

Table 10: Result of testing ANN when natural frequencies are used as features for transmission tower shown in Fig. 3.

Performance results of developed ANN models are significantly better when compared to other relevant results published in the literature for 2D structures [5, 1, 6, 7]. Furthermore effective ANN models are developed to predict damages in 3D structures with excellent performance results. Although ANNs are effective in detecting damage in the structure, the developed model can not be interpreted easily. Decision trees have the added benefit of generating rules that can be manually interpreted as illustrated in the case of transmission tower. Such rules may not be commonly known in the traditional analysis community and can be potentially useful to a structural designer.

The development of predictive model that can correctly predict the location and severity of damage in large complex structures can be a considerable challenge. For the case with variable loading and static displacements as features, the

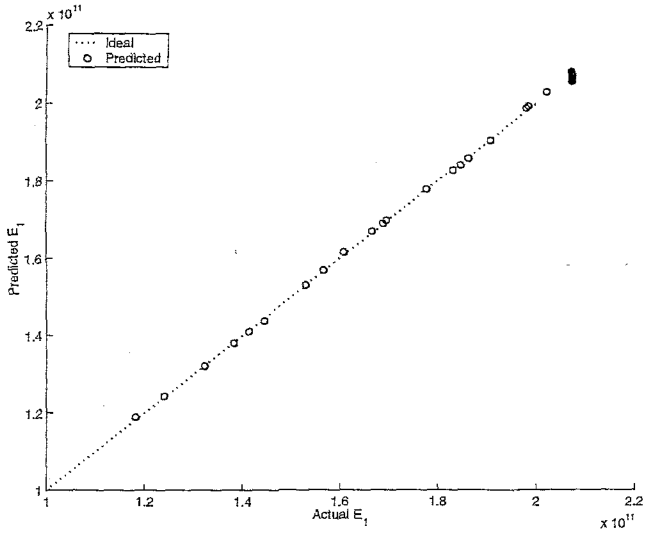


Figure 11: Comparison between predicted ideal and actual E for element 1 when natural frequencies are used as feature for electric transmission tower shown in Fig. 3.

models developed are not sufficiently accurate. Further work needs to be done to preprocess static displacements and extract features which will result in more accurate data mining models. Natural frequencies prove to be good features when load independent models are to be built. However, for complex structures, the values of natural frequencies are close to each other. This can cause the close natural frequencies to be mistaken for one another. To prevent this, MAC numbers (modal assurance criteria) can be used to distinguish such close frequencies, where MAC numbers are scalars that can distinguish two mode shapes from one another. Increased complexity of the structure would also cause the number of target variables (E), to increase. To handle this situation sub-structuring may need to be investigated.

6. REFERENCES

[1] Z.P. Szewczyk and Hajela. Damage detection in structures based on feature sensitive neural networks. *J. Comp. in Civ. Engrg., ASCE*, 8(2):163–179, 1994.

[2] X. Wu., J. Ghaboussi, and J.H. Garrett JR. Use of neural networks in detection of structural damage. *Computers and Structures*, 42(4):649–659, 1992.

[3] H. Chen and N. Bicanic. Assessment of Damage in Continuum Structures Based in Incomplete Modal Information. *Computers and Structures*, 74:559–570, 2000.

[4] J. V. A. Santos, C. M. M. Soares, C. A. M. Soares, and H. L. G. Pina. A Damage Identification Numerical Model Based on the Sensitivity of Orthogonality Conditions and Least Squares Techniques. *Computers and Structures*, 78:283–291, 2000.

[5] J. Zhao, J.N. Ivan, and J.T. DeWolf. Structural damage detection using artificial neural network. *J. of Infrastructure systems*, 4(3):93–101, 1998.

[6] G.J. Rix. Interpretation of nondestructive integrity tests using artificial neural networks. In *Struct. Congr. 12, ASCE, Reston, VA.*, pages 1246–1351, 1994.

[7] P. Tsou and M.H. Shen. Structural damage detection and identification using neural network. In *34th AIAA/ASME/ASCEAHS/ASC, Struct. Dyn. and Mat. Conf., AIAA/ASME Adaptive Struct. Forum, Pt.5*, 1993.

[8] D.E. Rumelhart and J.L. McClelland. *Parallel distributed processing: Explorations in the micro-structures of cognition.*, volume 1. MIT Press, Cambridge, Mass., 1986.

[9] J.T. DeWolf, P.E. Conn, and P.N. D'Leary. Continuous monitoring of bridge structures. In *IABSE Symp.*, 1995.

[10] M.S. Agbabian, S.F. Masri, M.I. Traina, and O. Waqfi. Detection of structural changes in a bridge model. In A.S.Novak, editor, *Bridge Evaluation, Repair and Rehabilitation, NATO Advanced Res. Workshop.* Kluwar Academic Publishers, Norwell, Mass, 1990.

[11] S.S. Law, H.S. Ward, G.B. Shi, R.Z. Chen, P. Waldron, and C. Taylor. Dynamic assessment of bridge load carrying capacities. *J. Strut. Engrg, ASCE*, 121(3):478–487, 1995.

Determination of an Initial Mesh Density for Finite Element Computations via Data Mining*

R. Kanapady, S. K. Bathina and K. K. Tamma
ramdev@me.umn.edu, sai@cs.umn.edu and ktamma@tc.umn.edu
Dept. of Mechanical Engineering, University of Minnesota, 111 Church St. S.E, Minneapolis MN 55455

C. Kamath
kamath2@llnl.gov
Center for Applied Scientific Computing, Lawrence Livermore National Laboratory, Livermore, CA 94551

V. Kumar
kumar@cs.umn.edu
Dept. of Computer Science and Engineering, University of Minnesota, 200 Union Street S.E, Minneapolis MN 55455

ABSTRACT

Numerical analysis software packages which employ a coarse first mesh or an inadequate initial mesh need to undergo a cumbersome and time consuming mesh refinement studies to obtain solutions with acceptable accuracy. Hence, it is critical for numerical methods such as finite element analysis to be able to determine a good initial mesh density for the subsequent finite element computations or as an input to a subsequent adaptive mesh generator. This paper explores the use of data mining techniques for obtaining an initial approximate finite element density that avoids significant trial and error to start finite element computations. As an illustration of proof of concept, a square plate which is simply supported at its edges and is subjected to a concentrated load is employed for the test case. Although simplistic, the present study provides insight into addressing the above considerations.

1. INTRODUCTION

It is widely recognized that the finite element method is the choice of many analysts for performing structural analysis simulations. It is a viable computational tool due to the various inherent advantages, namely, the capability of programming the method in a general purpose manner, the ability to handle natural boundary conditions and arbitrary loads acting on the structure, and the ability to model complex geometries. Various methods of generating finite element meshes exist in the literature. Some are based on prescribed mesh density values at various sample points in the geometry. Other approaches such as adaptive h, p, h-p refinements also exist. The so-called r-method of relocation of the nodes is yet another strategy for developing a suitable finite element mesh.

Numerical methods employing a coarse initial mesh suffer from the drawback of needing several successive mesh refinements for acceptable accuracy of results which tend to be cumbersome and expensive. It is well known that the procedures which start with a coarse mesh and attempt serious repetitive refinements, as is the case in most finite-element packages, are time consuming and costly. An approach of overcoming this limitation involves the use of some type of adaptive re-meshing scheme to guarantee convergence in the finite element solution. Whilst this approach is attractive, it can be slow to converge to ideal finite element meshes since the initial mesh for these adaptive schemes has zero knowledge of the problem apriori. Hence, close to ideal initial meshes of these adaptive re-meshing schemes may accelerate the convergence and guarantee sufficient accuracy in the finite element solution. Consequently this reduces the overall solution times for both serial and parallel architectures. Recent works [1] and [2] involved the application of Artificial Neural Networks (ANN) for the prediction of the finite element mesh density in order to estimate the magnetic field in a body. The present study builds upon previous work and provides a detailed study as related to structural mechanics applications. We specifically outline details in obtaining an ideal mesh densities at selected sampling points and an approach to enhance the quality of the results by asymmetric scaling of training samples.

In case of the structural mechanics, for illustration, Fig. 1 describes an elastic body Ω with boundary Γ which is de-

*The authors are very pleased to acknowledge support in part by the Department of Energy DOE/LLNL W-7045-ENG-48 and by the Army High Performance Computing Research Center (AHPCRC) under the auspices of the Department of the Army, Army Research Laboratory (ARL) cooperative agreement number DAAH04-95-2-0003/contract number DAAH04-95-C-0008. The content does not necessarily reflect the position or the policy of the government, and no official endorsement should be inferred. Access to computing facilities was provided by AHPCRC and Minnesota Supercomputer Institute

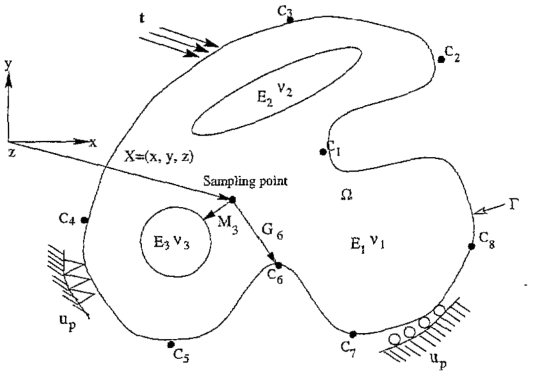


Figure 1: Illustrative application problem description for predicting finite element mesh density using data mining.

fined by representative critical points c ={c_1,c_2, ... } with respect to a fixed co-ordinate system. The elastic body is also comprised of materials with properties set M = {(E_1,v_1, ...),(E_2,v_2, ...), ... } where E_i, v_i are the Youngs modulus and the Poison's ratio respectively. The body is subjected to traction loads, **t**, and variety boundary constraints. The response of such a structure includes determination of field variables such as displacements, stresses and strain data for use in subsequent data mining models. Fig. 2 describes the finite element discretization with different material set M_i, loads t_i and boundary conditions u_{Pi}, i = 1,2, ... ,n where n is the number of finite element analyses carried out to generate the training data for the data mining model. From these analyses, it is postulated that one could predict an approximate mesh density for the analysis by employing error indicators formulated from data mining models. Hence for a given geometry, the objective is to predict the iniital finite element mesh density for arbitrary material distributions, loads, and boundary conditions as described in Fig. 3. Pictorially Fig. 2 describes the training examples to generate training data and Fig. 3 describes the test example for which the data mining model is required to predict the desired initial mesh density.

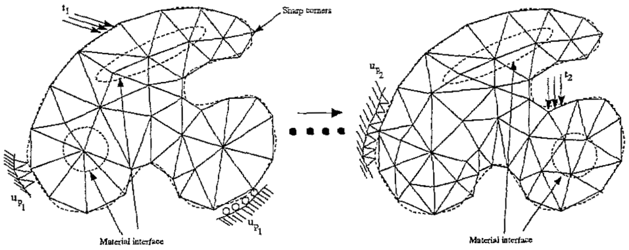


Figure 2: Illustrative application problem with finite element models for training example.

In this paper, as a proof-of-concept, we explore the calculation of the mesh density for a square plate, which is simply supported at its edges, with a concentrated load acting on it. This simplistic test example was selected since we already know the exact theoretical solution to the problem. From this, we can immediately assess if data mining techniques are indeed helpful for predicting the mesh density. The mesh density is predicted by training a simple feed forward neural network and making it learn the relationship between the mesh density and geometric features of the model. In Section 2, the preliminaries are discussed, followed by a discussion of the methodology used in predicting the mesh density in Section 3. In Section 4 and 5 the results obtained and conclusions of this study are discussed. In Section 6, future directions and the challenges involved are highlighted.

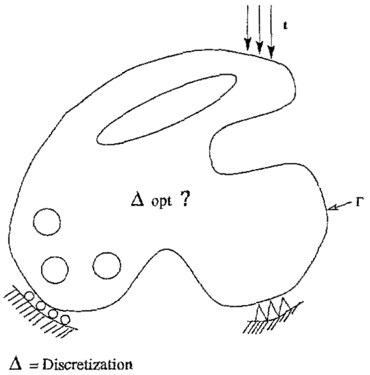


Figure 3: Overall goal of data mining illustrting application problem for predicting finite element mesh density.

2. PRELIMINARIES

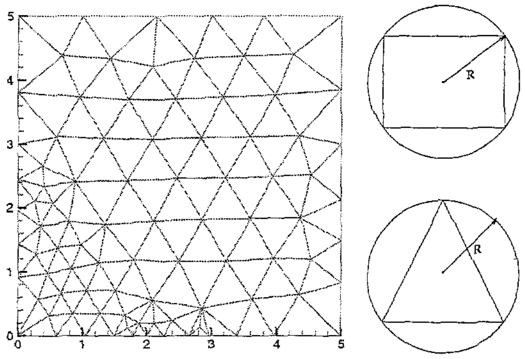
Finite element modeling involves discretizing the original domain into finite elements such as triangles. Such a typical process is shown in Fig. 4(a) using triangular elements for illustration, though other element types could also have been used. This is accomplished by a mesh generator for which the input is the mesh density at selected points in the domain. This mesh density can be defined in many ways. One such definition could be the number of nodes in the vicinity of a point [3]. Another definition could be the value of the radius (R) of the circle which is circumscribed over the triangle as shown in Fig. 4(b). This determines the triangle size and hence the element size in a finite element discretization. The mesh density value is the target variable of the classifier and the features can typically consist of geometry descriptions, loads applied, etc., depending on the problem at hand.

3. METHODOLOGY

In this section we discuss the various steps followed in calculating the initial mesh density for the problem at hand, the neural network architecture used to train the data, and feature selection required for training.

3.1 Generating the data

We start by training the predictive data mining models using example data, which pertains to "ideal" meshes of the representative geometries or domains. Here a square plate, is simply supported at its edges as shown in Fig. 5. A concentrated load, is applied at a point whose coordinates are (x_l,y_l). For this situation an analytical solution is available in [4], [5] which gives the displacement at any point (x_s,y_s)



(a) (b)

Figure 4: (a) Typical finite element mesh discretization and (b) ideal element representation for triangular and quadrilateral elements.

for any load at (x_l,y_l). The displacement w at (x_s,y_s) is given by

$$w = \frac{4P}{\pi^4 Dab} \sum_{m=1}^{\infty} \sum_{n=1}^{\infty} \frac{\sin(m\pi x_l)\sin(n\pi y_l)\sin(m\pi \frac{x_s}{a})\sin(n\pi \frac{y_s}{b})}{(\frac{x_s}{a})^2 + (\frac{y_s}{b})^2} \tag{1}$$

where P is the load applied at load coordinates (x_l, y_l), D is the flexural rigidity of the plate given by $\frac{Et^3}{12(1-v^2)}$, where E is the Young's modulus, t is the thickness, v is the Poisson's ratio, a is the length of the plate and b is the width of the plate. Once the displacement is obtained from this equation for a point (x_s,y_s), the mesh density value (h_{ideal}), which is the radius of the circle circumscribing the triangle (as shown in Fig. 4(b)), is determined for this point via the following steps:

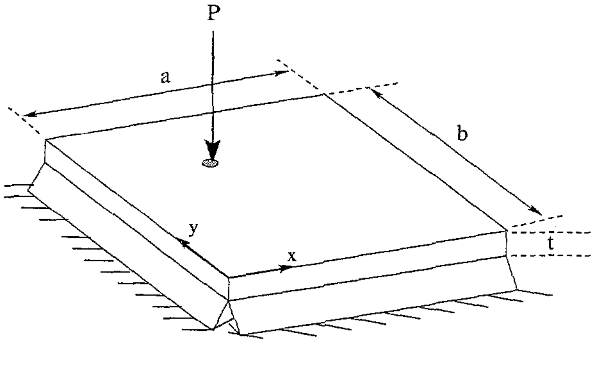
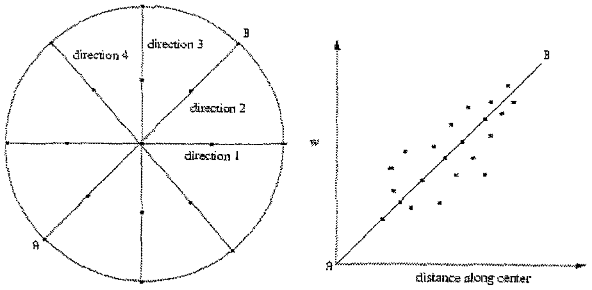


Figure 5: Problem description: a simply supported at all edges of square plate with concentrated load.

- First, choose a circular magnifier (radius of influence) of radius R_{ini} unit about the sample point (x_s,y_s). Next, start with and choose different directions in the circle at equal angular displacements. Then choose points along each direction such that they are equidistant from each other as shown in the Fig. 6(a). The displacements, w, are then computed at each of these points using Eq.(1).
- Next all the points that are on the line in a particular direction are chosen. The method of least squares is used to make the best linear fit as shown in Fig. 6(b) using various w's at the chosen points along this direction. The process is repeated for all the directions. Note that one could employ the best quadratic fit, best cubic fit, etc., depending on the application and the type of the finite element employed.
- The error is then estimated between the analytical solution and the numerical solution obtained from the best fit line, for each direction. Here, the error is defined by the L_2 norm on the solution vector in each direction. This norm should be less than the predefined tolerance limit, ϵ, for all the directions. Then, this value of the radius of the circle forms the h_{ideal} value, otherwise the radius is reduced. Note that the data mining model developed only holds for predetermined values of the tolerance limit, ϵ, for all other values the above steps have to be repeated till convergence to a numerical solution whose error lies within the tolerance limit.



(a) (b)

Figure 6: Ideal mesh density h_{ideal} computations for situation with known analytical response: (a) circular magnifier with points in chosen direction and (b) best fit line to exact values along a selected direction A-B.

Finally after obtaining the various h_{ideal} values at different sampling points in the domain, the mesh generator draws the approximate finite element mesh as shown Fig. 4(a).

3.2 Architecture of the Artificial Neural Network (ANN)

The artificial neural network considered in this work consists of a one input layer with 7 processing units corresponding one hidden layer with 19 processing units, and one output layer with a single processing unit. The ANN is trained to the corresponding target vector on the output layer. This

	Features							Target variable
Sampling points	Projxload	Projyload	ProjX	ProjY	d	P	t	h_{ideal}
1								
2								
⋮								
n								

Table 1: Training and test data layout for the problem.

target vector is the mesh density value for the finite element model.

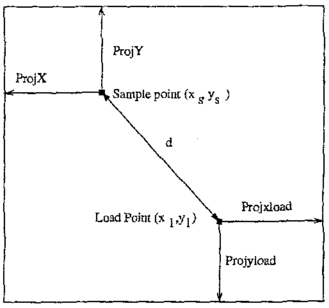


Figure 7: Features description for simply supported square plate with concentrated load.

3.3 Features selection

Since the model seeks to predict the element size, h_{ideal} at a point in the domain, the training data can have data sampled at many points in the domain of a single representative geometry. It is essential and critical that the features chosen for training have all of the characteristics that can be encountered in real applications such that the data-mining model becomes almost problem independent. The various features considered here for training the model are shown in Fig. 3.2 and include:

- Projxload, Projyload - projections of the load point (x_l,y_l) to the nearest adjacent edges,
- ProjX, ProjY - projections of sample point (x_s,y_s), where the mesh density is being determined, to the nearest adjacent edges,
- d - distance of the point (x_s,y_s) to the load point (x_l,y_l),
- P - load value,
- t - thickness of the plate.

For illustration, the corresponding feature table is described in Table 1.

4. RESULTS

Sample points were chosen randomly in the plate and the displacements were found for a load applied at different points which were again chosen randomly on the plate. The training data set consists of values of h_{ideal} in the range 0.01 to 1. This results in a finite element size scale ratio 1:100 which is the case in more realistic finite element applications. The distributions of h_{ideal} values are identical for training instances and testing instances.

load	0.68214
projpyload	0.56408
projpxload	0.50756
distance	0.32830
thickness	0.21520
projpx1	0.17636
projpy1	0.17247

Table 2: Reported relative importance of features to the developed neural network model.

The values of loads and thicknesses chosen for the testing case are different from the training case and, were chosen such that they are within the range of training case values. The training set consisted of 36,600 records and the test set consisted of 18,300 records. A neural net model was created using the training set. The relative importance of each of the features to the neural network using Clementine software [1] is listed in the Table 2. The plot between the predicted mesh value and the actual mesh value is shown in Fig. 8. In Fig. 8 the points above the diagonal represent predicted mesh sizes which are larger than the actual mesh size. This is detrimental for the finite element solution accuracy. Similarly, the points below the diagonal represent the predicted mesh sizes which are smaller than the actual mesh size. This is not detrimental for the the finite element solution. However, it is critical as the number of finite elements increases, it increases the computational cost. Restricting the attention only to the points above the diagonal, a scaling technique is used to bring down the points close to the diagonal by increasing the number of records pertaining to these points in the training set. The scaling technique used for this is given by

$$W = \frac{h^{pred}_{ideal} - h^{actual}_{ideal}}{h^{actual}_{ideal}} \times k \qquad (2)$$

where W is the number of records and k is a constant with a value of 2 in our case. Also, it is observed that most of these points pertain to the case where the load was close to the boundary of the plate. Therefore, in conjunction with the above scaling procedure load points that were within 0.5 units distance from the boundary of the plate were not considered in the new training set. A new neural net model was created with this new training set. The performance

[1] ©1999 SPSS Inc., Version 5.0.1

measures of the developed neural network model are listed in the Table 3. The plot between the predicted h_{ideal} and the actual h_{ideal} is shown in Fig. 9. Figures 10 – 13 show the finite element mesh generated employing these actual and predicted mesh densities for four loading conditions with 36 sampling points for each of them. The location of the loading point in each of the case is illustrated with a box on the plate. The actual mesh sizes for these four cases are obtained as mentioned in Section 3.1. The predicted mesh sizes are obtained from the neural network model. As the Figs. 10 – 13 show, the meshes for both the actual and predicted cases resemble each other very closely. As shown in table 4 the predicted number of mesh elements is very close to the actual number for all four cases.

Minimum Error	- 0.74468
Maximum Error	0.53632
Mean Error	0.0028778
Mean Absolute Error	0.019690
Standard deviation	0.051059
Linear Correlation	0.98256
Occurrences	18300

Table 3: Performance measures of the developed neural network model.

MESH	$\text{Elements}_{\text{Act}}$	$\text{Elements}_{\text{Pred}}$	% increase
Fig. 10	5667	5780	2.15
Fig. 11	7307	7414	1.46
Fig. 12	7270	7258	-0.17
Fig. 13	7318	7021	-4.06

Table 4: Actual and predicted number of elements for mesh in Figs. 10 – 13.

5. CONCLUDING REMARKS

The present work concentrated on a proof-of-concept application. A relatively simple problem where we know the theoretical solution was employed to assess the performance of data mining models. A simply supported square plate with a concentrated load was considered as a test case. Close to "ideal" mesh density h_{ideal} at various points in the plate were predicted with different load values, location and plate thickness. The training set was created without finite element discretization and this allows to create a data mining model. An ANN is employed and the initial results for predicting the appropriate mesh density are encouraging.

6. CHALLENGES AND FUTURE WORK

There are challenges involved in getting the training sets for a complex geometry subject to different loading conditions, composed of different materials and different boundary conditions (Ref. Figs. 1 – 3). Analytical solution are generally not known and we need to devise a strategy to determine mesh density values for a general case. Assessments of feasibility and performance are planned to be undertaken for various other data mining methods like Classification and Regression Trees (CART) and Multivariate adaptive regression splines (MARS) to find an alternative to neural network techniques. The overall goal is to create an effective system intended to provide an ideal initial mesh for a finite element simulation code or an initial "close to ideal" mesh for a subsequent adaptive solver employed for the finite element computations. Such a system will enable a knowledge-based approach for the pre-processing phase of finite element simulation codes.

7. REFERENCES

[1] D. N. Dyck, D. A. Lowther, and S. McFee. Determining an approximate finite element mesh density using neural network techniques. *IEEE Trans. Magn.*, 28, Mar 1992.

[2] C. H. Ahn. A self-organizing neural network approach for automatic mesh generation. *IEEE Trans. Magn.*, 27, 1990.

[3] R. Chedid and N. Najjar. Automatic finite-element mesh generation using artificial neural networks-part1: Prediction of mesh density. *IEEE Trans. Magn.*, 32(5), 1996.

[4] A. C. Ugural. *Stresses in Plates and Shells.* McGraw Hill, Inc, 1981.

[5] A. C. Ugural and D. Fenster. *Advanced applied stresses in Plates and Shells.* McGraw Hill, Inc, 1994.

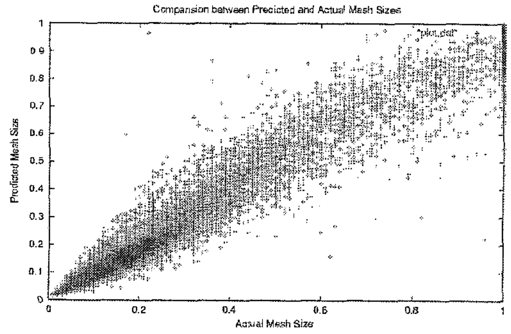


Figure 8: Comparison between predicted and actual mesh size before scaling.

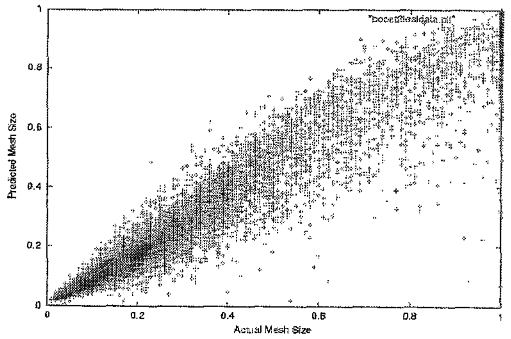
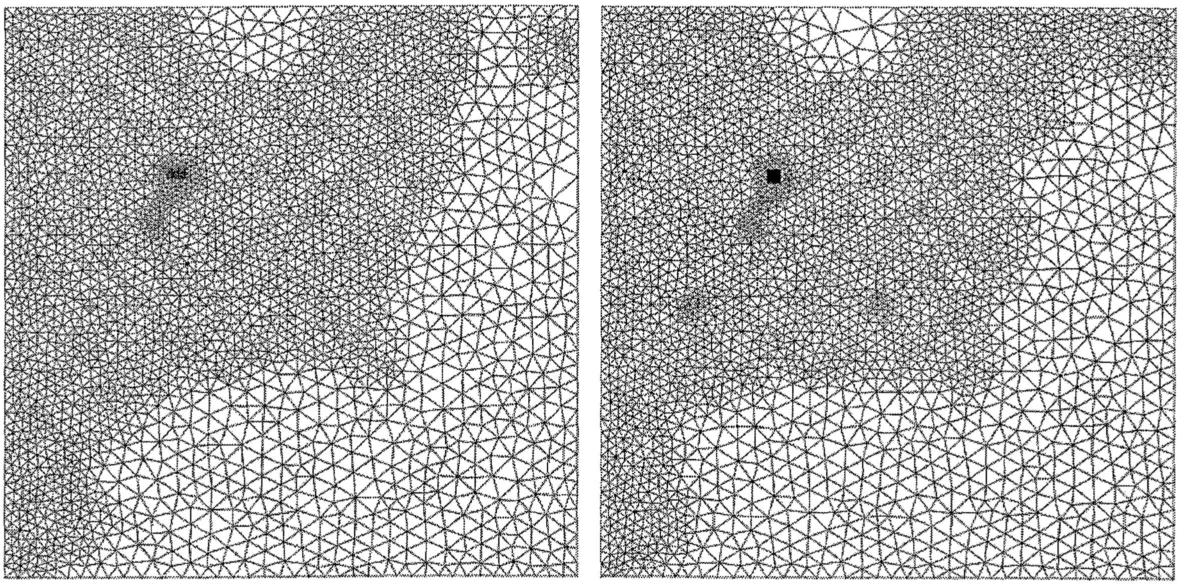


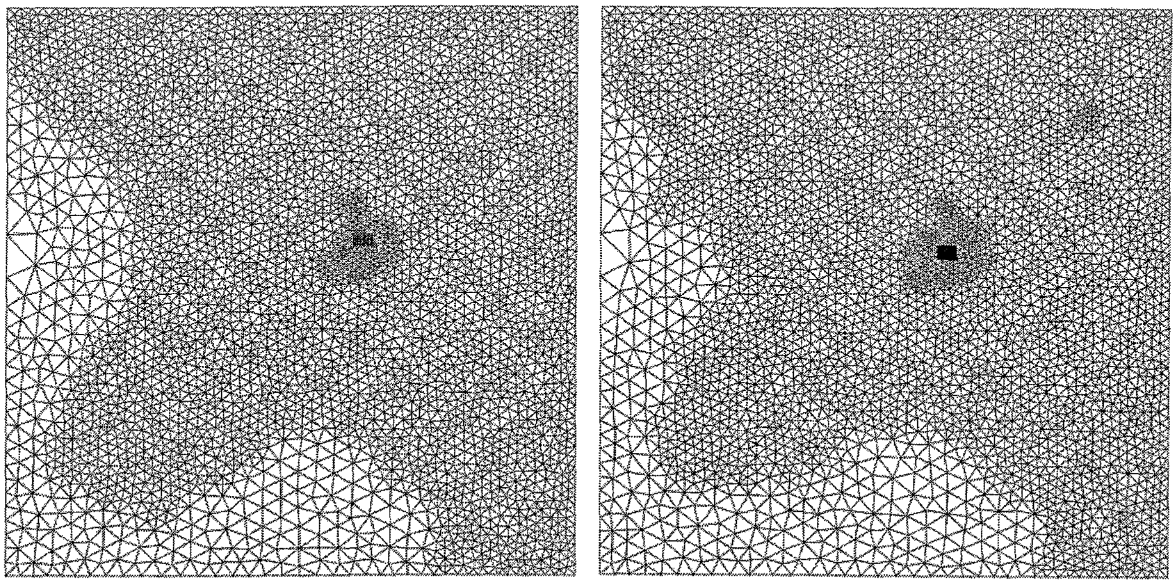
Figure 9: Comparison between predicted and actual mesh size after scaling.



(a) Actual mesh

(b) Predicted mesh

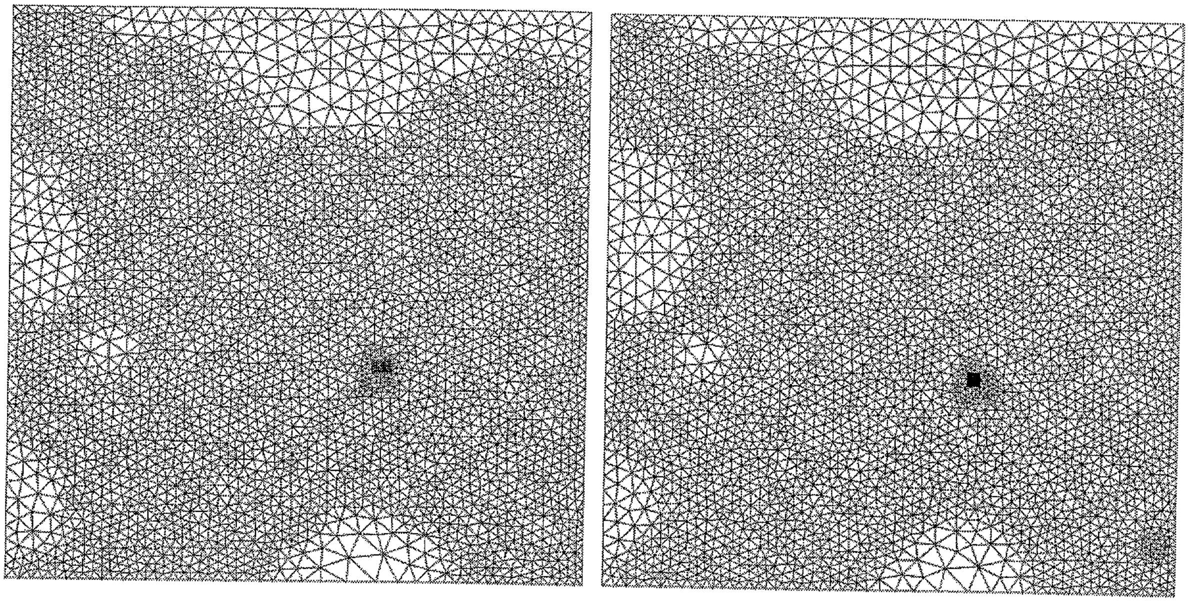
Figure 10: Finite element mesh for the load location (1.5,3.5).



(a) Actual mesh

(b) Predicted mesh

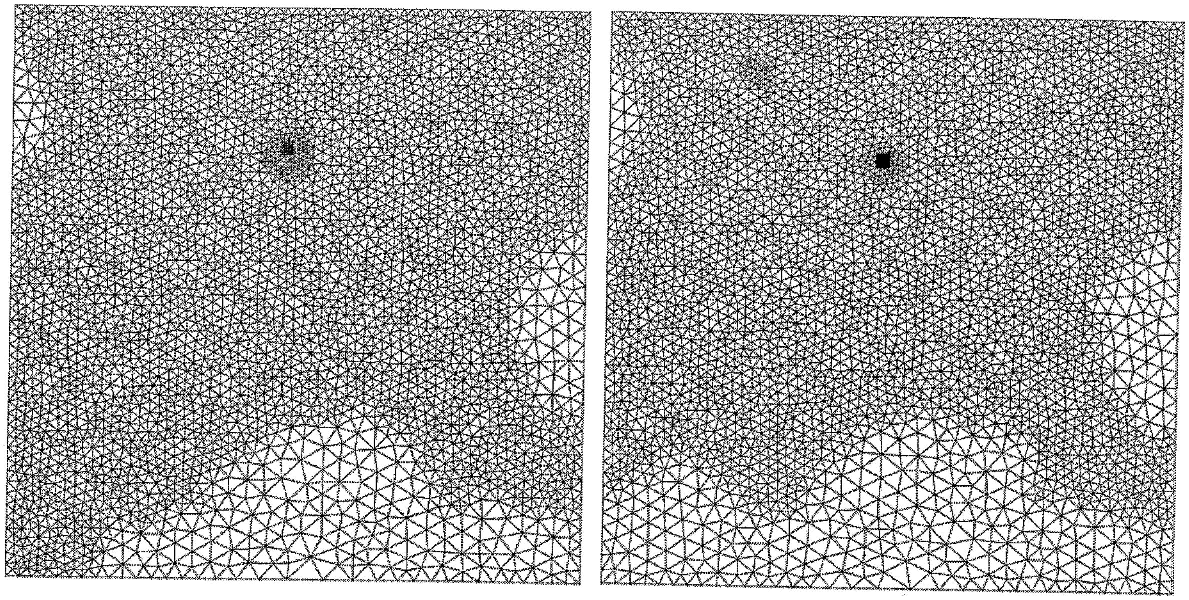
Figure 11: Finite element mesh for the load location (3.25,3).



(a) Actual mesh

(b) Predicted mesh

Figure 12: Finite element mesh for the load location (3.3,1.9).



(a) Actual mesh

(b) Predicted mesh

Figure 13: Finite element mesh for the load location (2.4,3.5).